Department of Computer Science
Series of Publications A
Report A-2020-10

Open Infrastructure for Edge Computing

Aleksandr Zavodovski

Doctoral dissertation, to be presented for public examination with the permission of the Faculty of Science of the University of Helsinki, in Auditorium CK112, Exactum building, on the 23rd of October, 2020 at 10 o'clock morning.

University of Helsinki
Finland

Supervisor
Jussi Kangasharju, University of Helsinki, Finland

Pre-examiners
Schahram Dustdar, TU Wien, Austria
Ming Zhao, Arizona State University, USA

Opponent
Mario Di Francesco, Aalto University, Finland

Custos
Jussi Kangasharju, University of Helsinki, Finland

Contact information

Department of Computer Science
P.O. Box 68 (Pietari Kalmin katu 5)
FI-00014 University of Helsinki
Finland

Email address: info@cs.helsinki.fi
URL: http://cs.helsinki.fi/
Telephone: +358 2941 911

ISSN 1238-8645
ISBN 978-951-51-6650-0 (paperback)
ISBN 978-951-51-6651-7 (PDF)
Helsinki 2020
Unigrafia

Open Infrastructure for Edge Computing

Aleksandr Zavodovski

Department of Computer Science
P.O. Box 68, FI-00014 University of Helsinki, Finland
aleksandr.zavodovski@helsinki.fi

PhD Thesis, Series of Publications A, Report A-2020-10
Helsinki, October 2020, 77+58 pages
ISSN 1238-8645
ISBN 978-951-51-6650-0 (paperback)
ISBN 978-951-51-6651-7 (PDF)

Abstract

Edge computing, bringing the computation closer to end-users and data producers, has now firmly gained the status of enabling technology for the new kinds of emerging applications, such as Virtual/Augmented Reality and Internet-of-Things (IoT). The motivation backing this rapidly developing computing paradigm is mainly two-fold. On the one hand, the goal is to minimize the latency that end-users experience, not only improving the quality of service but empowering new kinds of applications, which would not even be possible given higher delays. On the other, edge computing aims to save core networking bandwidth from being overwhelmed by myriads of IoT devices, sending their data to the cloud. After analyzing and aggregating IoT streams at edge servers, much less networking capacity will be required to persist remaining information in distant cloud datacenters.

Having a solid motivation and experiencing continuous interest from both academia and industry, edge computing is still in its nascency. To leave adolescence and take its place on a par with the cloud computing paradigm, finally forming a versatile edge-cloud environment, the newcomer needs to overcome a number of challenges. First of all, the computing infrastructure to deploy edge applications and services is very limited at the moment. Indeed, there are initiatives supported by the telecommunication industry, like Multi-access Edge Computing (MEC). Also, Cloud Service Providers (CSPs) plan to establish their facilities near the edge of the network. However, we believe that even more efforts will be required to make edge servers generally available. Second, to emerge and function efficiently, the ecosys-

tem of edge computing needs practices, standards, and governance mechanisms of its own kind. The specificity originates from the highly dispersed nature of the edge, implying high heterogeneity of resources and diverse administrative control over the computing facilities. Finally, the third challenge is the dynamicity of the edge computing environment due to, e.g., varying demand, migrating clients, etc.

In this thesis, we outline underlying principles of what we call an Open Infrastructure for Edge (OpenIE), identify its key features, and provide solutions for them. Intended to tackle the challenges we mentioned above, OpenIE defines a set of common practices and loosely coupled technologies creating a unified environment out of highly heterogeneous and administratively partitioned edge computing resources. Particularly, we design a protocol capable of discovering edge providers on a global scale. Further, we propose a framework of Intelligent Containers (ICONs), capable of autonomous decision making and forming a service overlay on a large-scale edge-cloud setting. As edge providers need to be economically incentivized, we devise a Dominant Strategy Incentive Compatible (DSIC) double auction mechanism where edge providers can meet application owners or administrators in need of deploying an edge service. Due to DSIC property, in our auction, it is the best strategy for all participants to bid one's privately known valuation (or cost), thus making complex market behavior strategies obsolete. We analyze the potential of Distributed Ledger Technologies (DLT) to serve for OpenIE decentralized agreement and transaction handling and show how our auction can be implemented with the help of DLT. With the key building blocks of OpenIE mentioned above, we hope to make an entrance for anyone interested in service provisioning at the edge as easy as possible. We hope that with the emergence of Independent Edge Providers (IEPs), edge computing will finally become pervasive.

Computing Reviews (2012) Categories and Subject Descriptors:

Computer systems organization → Distributed architectures
Networks → Network services
Networks → Network performance evaluation

General Terms:
Design, Experimentation, Performance, Protocols, Platforms, Incentive Compatibility, Game theory, Auction theory, Mechanism design, Smart contract, Blockchain, Distributed Ledger

Additional Key Words and Phrases:
Edge computing, Task deployment, Virtualization

Acknowledgements

I wish to express my warmest gratitude to my supervisor, Prof. Jussi Kangasharju, whose optimism and confidence guided me through this sometimes thorny path. The vitalizing atmosphere of exploration and discovery he has managed to create in his lab will always accompany me. My sincere thankfulness goes to my co-supervisor, Prof. Suzan Bayhan, with whom my scientific career started at the time of writing a Master thesis. Her ever encouraging, relentless support continued over my Ph.D. studies, invaluably contributing to my progress in the challenging field of science.

I'm happy to convey my acknowledgments to Prof. Emeritus Rauli Svento, whose inspiring attitude and wisdom have always encouraged me throughout the rollercoaster of rejects and accepts. I also thank all the members of the BCDC project, with whom I have learned a lot and shared many remarkable moments. This work would not have been possible without the funding and comprehensive support provided by the BCDC project and the Doctoral Programme in Computer Science (DoCS).

I am much obliged to the pre-examiners, Prof. Schahram Dustdar and Prof. Ming Zhao, who benevolently took so much effort in improving this thesis to the highest possible quality standards. I express my deep gratitude to Prof. Mario Di Francesco for becoming my opponent.

I'm highly obliged to our lab members: Nitinder Mohan, Otto Waltari, Walter Wong, Pengyuan Zhou, and our special guest Lorenzo Corneo. Besides being great contributors and mentors, they have created such an enlightened and friendly ambience in our lab, that I wish my Ph.D. journey would continue longer.

I owe my earnest appreciation to the teachers and all the staff of the Department of Computer Science at the University of Helsinki, my Alma Mater, where I began my studies as a Master student. I thank Pirjo Moen for her invaluable help with overcoming the administrative hurdles.

Finally, I would like to thank my family for their patience, love and support during this remarkable period of my life.

In Helsinki, Finland, October, 2020
Aleksandr Zavodovski

List of Abbreviations

AP Access Point. 39, 40
AR Augmented Reality. 2
AR/VR Augmented/Virtual Reality. 2, 16, 17
AWS Amazon Web Services. 4, 6, 7, 20, 38, 48
BOINC Berkeley Open Infrastructure for Network Computing. 4, 5, 24, 27
CAIDA Center for Applied Internet Data Analysis. 31, 34, 38
CDN Content Delivery Network. 1, 25, 26
CEC Crowdsourced Edge Computing. 25
CSP Cloud Service Provider. iii, 3–5, 7–9, 19–21, 26, 28, 30, 39
DC/OS Distributed Cloud Operating System. 6, 20, 32
DLT Distributed Ledger Technologies. iv, xi, xii, 5, 10, 11, 13, 15, 22–24, 27, 29, 33, 41, 47–51, 54, 55
DNS Domain Name System. 31, 32, 53, 55
DPoS Delegated Proof-of-Stake. 49
DSIC Dominant Strategy Incentive Compatible. iv, 25, 40–42, 44–46, 54
ETSI European Telecommunications Standards Institute. 4, 18, 20
ExEC Elastic Extensible Cloud. 13, 30–34, 36, 53–55
HTR Human Reaction Time. 16, 17
ICON Intelligent Container. iv, xi, 10, 13, 29, 35–39, 48, 54, 55
IEEE Institute of Electrical and Electronics Engineers. 22
IEP Independent Edge Provider. iv, xi, 9, 10, 30–40, 48, 50, 53, 54
IIC Industrial Internet Consortium. 21, 22
IoT Internet-of-Things. iii, xi, 1, 2, 8, 16, 17, 21–23, 25
IP Internet Protocol. 30, 31
LoRa Long Range. 21
LPWAN Low-Power Wide-Area Network. 21
MANO Management and Network Orchestration. 19
MCC Mobile Cloud Computing. 19
MEC Multi-access Edge Computing. iii, 4, 8, 9, 17–20, 30
ML Machine Learning. 26

MNO Mobile Network Operator. 8, 9, 18, 20, 30, 39
MTP Motion-to-Photon. 16, 17
NASA National Aeronautics and Space Administration. 16
NFV Network Function Virtualization. 19
OEC Open Edge Computing Initiative. 20, 22
OpenIE Open Infrastructure for Edge. iv, xi, xii, 9–11, 13, 15, 24, 26, 29, 30, 33, 40, 41, 43, 45, 47–51, 53–55
P2P Peer-to-Peer. 23, 41, 43
PL Perceivable Latency. 16, 17
PoS Proof-of-Stake. 26, 43, 49, 50, 54
PoW Proof-of-Work. 22, 43, 49, 54
QoE Quality-of-Experience. 18, 33
QoS Quality-of-Service. 1, 25
SDN Software-Defined Networking. 19
SLA Service Level Agreement. 8, 24, 33, 48
TEE Trusted Execution Environment. 9, 30
UAV Unmanned Aerial Vehicle. 2
VM Virtual Machine. 6, 19, 30, 48
WASM WebAssembly. 26

Contents

Chapter 1

Introduction

The term *edge*, as referred to as a part of a network close to end-users, has made its first appearance in the context of Content Delivery Networks (CDNs) [44]. At the beginning of the current millennium, CDNs started to deploy servers near their clients with the purpose of improving the quality of content delivery. Compared to a centralized content server, such a solution offered not only an improved Quality-of-Service (QoS), but also substantial bandwidth savings. Those servers, functioning as mere caches for content, became known as *edge servers*, according to their location in the topology of a network. The rest of the first decade of the 3rd millennium was dominated by the rise of cloud computing, resulting in the appearance of centralized, large-scale datacenter deployments. However, these were separated from most of the end-users by networking delay ranging from seconds to hundreds of milliseconds, as Li et al. [119] pointed out in 2010.

Edge re-entered the scene in 2009, as Satyanarayanan et al. [167] suggested cloudlets – computing facilities near end-users. The rationale was that mobile devices, which were getting intrinsically sophisticated, could offload complex computational tasks to cloudlets for execution. These local compute boxes or even small datacenters were envisioned to empower applications beyond hardware limitations of handheld gadgets. In contrast to CDNs, not storage but the computation becomes the main driving force. Cloud datacenters could not handle offloading since they were too distant to satisfy delay requirements imposed by the interactive nature of mobile applications.

Meanwhile, another technological trend was becoming increasingly prominent, namely, the Internet-of-Things (IoT) [59]. The IoT trend continues strong at present, and the recent Cisco report suggests over 28 billion connected devices by the end of 2022 [35]. The amount of data that such devices generate has been exponentially increasing during the past few years [64].

The need for local data processing for IoT was realized early, emanating in *fog computing* proposed by Bonomi et al. [21] as a solution. Essentially, edge and fog share a lot of similar concepts. Therefore, the terms are quite often used interchangeably, with fog having some sort of IoT flavor, intended mainly to operate on upstream data, while edge historically refers to reducing downstream latency that clients experience. In this work, for clarity purposes, we will use edge computing as an umbrella term encompassing all kinds of computation performed at the edge of the network, including IoT data processing and aggregation.

Receiving strong initial impetus from cloudlets, edge computing gained a lot of additional momentum from IoT, and at present, firmly holds the status of enabling technology for the new kinds of emerging applications, such as Augmented/Virtual Reality (AR/VR) [55, 122, 172], smart cities [34, 74, 162], connected vehicles [104, 217], extending the capabilities of Unmanned Aerial Vehicles (UAVs) [24], and many others. To illustrate, edge computing can enable AR city tours, where customers wearing special glasses can get additional information about the objects they observe [183], while videos and other shareable content produced during such a tour can be cached by near-by edge servers. Potentially dangerous for vehicular traffic, black ice road conditions can be detected more efficiently in a collaborative fashion. Edge servers can aggregate and process sensor data is on a regional basis, with the analysis results subsequently being available to all traffic participants in the area [41]. Continuing the topic, improvements that edge computing has to offer for road safety are diverse and abundant, ranging from driving assistance [88] to completely autonomous self-driving vehicles [120, 148]. Also, optimizing transportation by controlling traffic lights has been considered [216]. Edge computing is essential for enabling complex industrial robotics environments [29, 94, 140], as near-by edge offers computational capacity and coordination that robots require. The upstream aggregator and preprocessor role of edge computing in IoT has recently been augmented by distributed analytics capabilities [30,32,163,169], further reducing the load on the network and cloud datacenters.

Given diverse areas where edge computing is becoming increasingly relevant, is it rather difficult to distinguish what might become a "killer application"? However, Ananthanarayanan et al. [5] suggest that real-time video analytics is the most likely candidate to claim such a status. The previously mentioned use cases and scenarios to which edge computing is applicable are by no means a complete list, and numerous comprehensive surveys, position papers, and case studies have to offer much more on the topic, e.g., [63, 73, 89, 129, 153, 159, 168, 175].

1.1 Background and Motivation

With a brief overview of the versatile pragmatic scenarios presented above, it is clear that the demand for edge computing is rather strong and increasing. Next, we proceed to examine the current condition of this novel emerging paradigm. We refer to a set of software platforms and components, protocols, and common practices essential for edge computing as *virtual infrastructure*. Likewise, we use the term *physical infrastructure* for servers and other devices used at the edge to perform computation. Together, physical and virtual infrastructures form an *ecosystem* of edge computing, which can be characterized by considering the following questions:

1. What is the present state of physical infrastructure, i.e., are there any edge servers where interested parties can deploy their applications?
2. What has been accomplished in the virtual infrastructure, i.e., what platforms, protocols, etc. suitable for edge computing exist? How satisfactory are those solutions, and what deficiencies do they have?
3. How are initiatives of key industry players, open-source communities, academic influencers likely to affect the future ecosystem of edge computing?

We discuss the questions related to physical and virtual infrastructure in the next two sections accordingly. The discussion of the last question, intertwining with the outlook for the ecosystem of edge computing in general, will complement both of the sections.

1.1.1 Physical Infrastructure

When it comes to physical infrastructure and the general availability of edge servers, we can confidently conclude that it is very limited at the moment of writing. Contrasting with well-established Cloud Service Providers (CSPs), there are no comparable edge service providers currently in the business. While lacking general-purpose edge servers, ones in need of those are offered to buy edge computing devices and related gear themselves [65]. However, there are initiatives by prominent industry players and various consortia to establish a generally available physical infrastructure.

First, there is Multi-access Edge Computing (MEC)[1] proposed and actively developed by the European Telecommunications Standards Institute

[1]MEC was previously known as Mobile Edge Computing, and became Multi-access Edge Computing later due to added WiFi and fixed connections support.

(ETSI) [58]. As its origin implies, MEC is telecom-oriented, and practical installations are expected to begin along with 5G network deployments, although the technology itself neither requires 5G nor depends on it. Currently, to the best of our knowledge, MEC exists only in the form of pilots, trials, or proof-of-concept projects, e.g., [25]. So far, it remains uncertain, whether telecom operators will undertake the financial risks of large-scale MEC deployment in the future.

Second, CSPs are extending their infrastructure to the edge. Microsoft has a clearly defined edge strategy [194], of which the key component is Azure Stack [135]. Amazon Web Services (AWS) and Google are moving in a similar direction, offering their own edge IoT ecosystems, [10,11] and [81], respectively. In the case of IoT services provisioned by major CSPs, there are no local edge servers on behalf of the cloud provider, and customers have to acquire, deploy, and support compatible devices themselves.[2] To surpass this limitation, Google seeks to opt for cooperation with telecom operators for physical deployments of edge servers [117]; however, relying mostly on its own set of technologies instead of MEC. AWS approaches the problem by introducing *local zones* [49]. The initial intention is to establish datacenters in densely populated areas, providing the majority of the population with the benefits of edge computing, e.g., low latency service. Similarly to Google, AWS also considers cooperation with telecom providers, planning to make their local zone facilities easily accessible from the future 5G networks. Microsoft's future cloud strategy is also tightly coupled with the edge, but it is not yet much concretized when it comes to practical deployments [136]. Nevertheless, Azure's world-wide regional coverage is subject to further expansion, as can be deduced from numerous new planned datacenter locations [134].

Third, there is an opportunity for crowdsourcing. Indeed, in scientific computing, crowdsourcing has been around quite a long time, with the best likely known example being Berkeley Open Infrastructure for Network Computing (BOINC) platform [6], which became widely recognizable due to the SETI@home project.[3] The factor that makes crowdsourcing particularly attractive for edge computing is that consumers devices are in abundance at the edge on the network, exactly where computing capacity is needed the most. Moreover, the hardware of those devices is mostly underutilized. The recent uptrend in Distributed Ledger Technologies (DLT), promoting possibilities for distributed agreement via smart

[2] Users of Azure Stack and AWS Outposts can order ready-to-operate devices directly from Microsoft and Amazon, respectively.

[3] BOINC is specifically designed for scientific computing and is not suitable to perform ad hoc computational tasks at the edge.

contracts [181, 196], and enabling economic incentives in the form of cryptographic tokens (e.g., Bitcoin [142]), brightens the perspectives for crowdsourcing further. Expectedly, a number of new crowdsourcing platforms built on top of DLT appeared, e.g., iExec [100], Sonm [179], Golem [79], Edge Network [51], just to name a few. While some position themselves as an alternative to conventional CSPs (e.g., Golem and iExec), others have a distinct edge focus (e.g., Sonm and Edge Network). However, newcomers have severe difficulties in reaching the level of BOINC in popularity. At the moment of writing, the Golem network had 157 nodes [78], and Edge Network [52] – 659, compared to 141 417 volunteers and 835 414 computers available on BOINC [20]. The crowdsourcing platforms of today do not reach the level required to establish edge computing as a pervasive paradigm. Still, the success of BOINC and perspectives brought by DLT keep the possibility open. Moreover, there are already systematic attempts to define contours of the future crowdsourced edge frameworks, e.g., [116].

The main challenge for the physical infrastructure is two-fold. On the one hand, to establish edge computing accomplished paradigm deployments need to be pervasive covering vast geographical regions, i.e., edge servers must be close to clients everywhere. On the other, the need to cover large geographical areas with server deployments deprives edge computing of quite an important success factor behind the rise of the cloud datacenters, namely, the economy of scale [193], when grouping vast amounts of computing equipment together on a single site leads to remarkable monetary savings in maintenance, procurement, and other support activities. In the case of edge computing, this is not quite achievable. Given the present situation, it is not yet clear how the impediments will be circumvented, thus posing a research opportunity.

1.1.2 Virtual Infrastructure

As we already mentioned, under the term virtual infrastructure, we group platforms, technical software solutions, protocols, and common practices aimed to facilitate edge computing. In contrast to the previously discussed physical deployments, much more is achieved here, probably due to the less economically demanding nature of the tasks, as compared to maintaining vast installations of edge servers. The endeavor comes from both academy and industry, often resulting in mature, ready-to-operate building blocks or entire platforms. Since the edge is coming to the cloud-dominant world [137], many solutions, which have proven themselves efficient in the context of cloud computing, are currently being extended to the edge.

Virtualization [68, 199] has been empowering the cloud since its early beginning and has greatly contributed to the popularity of microservices architecture in general [107]. Most of these technologies are directly utilizable also at the edge, e.g., Docker [48] containers perform fine in this new context [105]. The same applies to LXC [70] Linux containers [152]. Virtual Machines (VMs), which sometimes are preferable for stateful applications, are also considered by many as a viable choice [1,168,211]. Mobility use-cases are intrinsic for edge computing, and migration of VMs or containers is also profoundly explored in [27, 124]. Moreover, the constrained capabilities of edge devices, which may put limitations on the support of conventional VMs and containers, are taken into account by considering the usage of MirageOS unikernels [40, 101, 141]. Addressing the same problem, AWS has developed Firecracker [17] – a lightweight virtualization framework supporting microVMs[4], which in comparison with conventional VMs have much lower memory impact and are faster to instantiate.

On the level of management and orchestration, we also witness the transferability of cloud technology solutions to edge computing. Kubernetes [146] – the de facto standard in container orchestration [75], is now reaching out to the edge with its KubeEdge spinoff project [145]. Distributed Cloud Operating System (DC/OS) [43, 205], which is a result of the commercial development of Mesos cluster manager [66, 91] by D2iQ[5], is now actively supporting edge computing [53], remarkably, even for fairly advanced disconnected scenarios [106]. OpenNebula is an open-sourced cloud platform [150] capable of hybrid edge-cloud environment governance. Similarly, OpenStack [151] highly prioritizes the edge computing support, and there is already a cloudlet-oriented flavor, namely, OpenStack++ [87]. The above highlights are only the tip of the iceberg, as they were preceded by a substantial academic body of work and other industry-led efforts. Since the conception of edge, either in cloudlets [167] or fog [21], its relation to the cloud was constitutional, subsequently evolving into *edge-cloud* environments of various kinds, e.g., [26, 50, 125, 138, 177, 191].

The serverless trend in cloud computing [14, 110], already taken up by major vendors with Azure Functions [133] and AWS Lambda [12] being two of the most prominent examples, is likewise applicable for edge computing. The attractiveness of the serverless approach is that an application is automatically mapped to required compute resources, abstracting from the underlying execution infrastructure. The already mentioned AWS Firecracker [17] is a serverless platform. Numerous academically developed

[4] Also called kernel-based VMs [68].

[5] Formerly known as Mesosphere Inc.

platforms demonstrate the potential of the serverless approach for edge computing, e.g., [9, 15, 16, 45, 77, 144, 164, 204].

As our short glimpse at edge solutions and platforms suggests, the problem with the virtual edge infrastructure appears to be siloing and lack of interoperability. Physical computing devices at the edge might be scarce. Nevertheless, CSPs generally assume that edge servers at their disposal will function seamlessly only with cloud back-end they also host, but even in the case of underutilization, it will not be possible to host edge loads from other competitive CSPs, not to mention other use cases. The cause of the problem is the lack of appropriate virtual infrastructure in a broad sense, that would provide interoperability fabric and accompanying services, such as discovery, agreement, financial transaction handling, etc. Moreover, such virtual infrastructure would make it encouraging for all interested parties to become edge service providers, increasing the pervasiveness of the edge and boosting its evolution into a full-fledged computing paradigm.

1.2 Problem Statement

To succeed, edge computing needs to be pervasive. Due to its purpose, i.e., bringing computation closer to clients, the nature of required edge deployments drastically diverts from what we observe in the case of the cloud: a plethora of small geographically scattered installations versus a few big, capable datacenters. Based on our previously presented viewpoints, we can roughly identify two major challenges that edge computing is currently facing: lack of general-purpose edge servers and shortcomings of virtual infrastructure. These two are closely interrelated, and lack of appropriate platforms and practices can slow down progress in physical infrastructure deployment, as we show next.

At the time being, the establishment of cloud datacenters required enormous investments, which only big corporations could afford. The economy of scale made returns of those investments spectacularly high. If we consider the edge, to cover an area of a limited size and population relatively low investments are required, as a few capable machines will likely to be sufficient for serving near-by clients well in most cases. The problem is where the motivation for those local edge service providers will come from? Clearly, they will neither benefit from the economy of scale nor offer the elasticity of the cloud to their clients. Logically, to be cost-effective, those small providers need to serve as many potential clients as possible, hosting any edge application that those clients might require. Due to the mobility of clients or changes in IoT deployments, the list of such applications may

vary dynamically. Another concern is that edge providers must be located promptly by clients who are using edge services and, likewise, any party that is interested in deploying applications to the edge. If the client is mobile and migrates from one area to another, it is convenient that the edge service follows such a client. In case the edge providers in those areas are different, e.g., one is a MEC installation, and the other is hosted by CSP, we face an *edge-to-edge* interoperability challenge. Alternatively, cross-provider *cloud-to-edge* interoperability is desirable in a situation, where the CSP currently hosting both edge and cloud components of an application does not have any edge facility in a new area where demand for the application grows. In such a case, it is desirable that the CSP could locate and utilize edge facilities administrated by other parties, e.g., competitive CSPs, Mobile Network Operators (MNOs), or crowdsourced providers.

Next, we present a systematic view on what a virtual infrastructure would need to facilitate the emergence of third-party edge providers and improve the interoperability between existing and upcoming ones, e.g., MEC, cloud, or crowdsourcing initiatives. The following features are essential:

F1. **Discovery mechanism** for computational resources at the edge, capable of operating on a global scale and built on top of already deployed protocols and networking infrastructure.

F2. **Negotiation protocol** for querying edge facilities about available time slots for service deployment, available hardware, networking resources, price, and financial transaction handling preferences.

F3. **Autonomous adaptation** to handle the dynamic challenges of edge computing and reduce administrative effort. For instance, edge services should migrate following their clients and discover the required deployment facilities autonomously.

F4. **The economic model** for efficient trade between edge providers and those who wish to deploy their applications on edge facilities.

F5. **Digital agreement** as the edge providers might need to negotiate and digitally sign Service Level Agreement (SLA)-like treaties.

F6. **Financial transactions' handling**, as the edge providers need an economical incentive, i.e., monetary compensation for their services.

We will call a virtual infrastructure encompassing features enumerated above – Open Infrastructure for Edge (OpenIE), and edge providers supporting OpenIE – Independent Edge Providers (IEPs). Although the rationale behind OpenIE is to enable the entrance of third-party providers,

e.g., administrative entities who possess some local datacenter, or individuals exposing their devices via crowdsourcing, we are not ruling out CSPs or the MEC initiative of MNOs. As long as their computing facilities will conform to OpenIE practices, they can also be treated as any other IEP and operate in a similar fashion. In any case, supporting OpenIE will only increase the chances of higher utilization of edge computing facilities. Moreover, anyone compliant with OpenIE will have chances to provide better service coverage for their clients since not only their own edge facilities will be available but also those provided by other IEPs.

OpenIE might make an impression of too problematic undertaking from the technical perspective and, thus, unpractical to implement. Truly, there are challenges, but as we already have shown, the vast technology stack that edge computing inherits from the cloud, along with numerous specifically developed platforms and protocols give reasons for cautious optimism. Security and trust issues are of paramount importance when it comes to open environments. Trusted Execution Environments (TEEs) have the potential to mitigate the trust issues that may arise between IEP and application owners. Memory regions protected by TEE, i.e., enclaves are not accessible even by users with root privileges. TEEs are on solid ground in the industry, and all major processor manufacturers currently support them [8,103]. Further, protecting information with enclaves in edge and cloud computing is already explored in [86,185].

OpenIE is affecting not only edge computing but once a reality, it can make a drastic impact on the cloud itself. On the one hand, the idea of multi-cloud environments, where multiple CSPs are acting as a unified logical cloud, has been around for a long time [2,157,176]. On the other, the present state of the Internet is characterized as suffering from ossification and monopolization [121]. Sharing the previous arguments, Peterson et al. appeal for the democratization of the network edge [158]. The entrance of third-party providers can not only make edge computing a reality, but also dramatically reduce monopolization in cloud service provisioning.

The establishment of OpenIE in its entirety is clearly out of this thesis scope. Therefore, we limit the exploration of features F1 – F6 by posing the following research questions:

RQ1. *Is a global-scale discovery mechanism for edge resources feasible, and how should it operate?*

RQ2. *How can autonomous decision making tame the dynamicity required of edge computing, and how can it be incorporated into practical systems?*

RQ3. *How will the presence of IEPs affect the placement of edge servers?*

RQ4. *What market model would be beneficial for edge providers and application owners to enter into the trade, and how can such a market be technically implemented?*

RQ5. *What potential do existing DLT platforms have to provide agreement and transaction handling services for OpenIE?*

1.3 Thesis Contributions

The major contribution of this thesis is an attempt to bring edge computing closer towards becoming a pervasive full-fledged computing paradigm, which would organically complement the cloud. The endeavor resulted in the suggestion of Open Infrastructure for Edge (OpenIE) and identification of its key features. The set of platforms, protocols, and common practices forming the core of OpenIE is intended to facilitate cross-provider cloud-to-edge and edge-to-edge interoperability. We hope that OpenIE will also promote the emergence of a new kind of edge service providers, namely, Independent Edge Providers (IEPs). These might be business entities or individuals either possessing underutilized computational devices at the edge or interested in establishing a new computing facility. With the OpenIE initiative, we aim at creating a unified environment, set of best practices, and standards for moving and deploying services across the network. In this thesis, the models, technical solutions, and evaluations for the following foundational features of OpenIE are presented:

- The discovery protocol, capable of locating edge computing resources at global-scale. The solution relies only on existent networking infrastructure and requires no changes to it.

- The central building block of OpenIE – Intelligent Container (ICON). Capable of autonomous decision making, ICON uses the previously mentioned discovery protocol, and once an IEP near clients is found, it either migrates or deploys its replica to this new location, optimizing the service delivery. An ICON continuously monitors the behavior of the clients and reacts promptly, tackling the dynamic challenges of the edge environment.

- The market model – truthful double auction where clients and providers can submit their bids to buy and sell computational resources at the

edge. The optimal strategy in truthful auctions is to submit a privately known valuation (or cost in the case of a provider) to the auctioneer, thus, eliminating the price manipulation attempts along with the need for complex strategizing.

- The analysis of existing DLT platforms potential to serve OpenIE for agreement and financial transaction handling. Our conclusion is semi-positive since, as of now, there are non-negligible shortcomings in all major DLT platforms, but the development in the field is promising. We show how the previously presented market model can be implemented on top of a DLT in distributed fashion, as an auction without auctioneer.

This thesis incorporates the research published in the following five original articles and a manuscript under submission.

Publication I: ExEC: Elastic Extensible Edge Cloud. Aleksandr Zavodovski, Nitinder Mohan, Suzan Bayhan, Walter Wong, and Jussi Kangasharju. Published in ACM Workshop on Edge Systems, Analytics and Networking (EdgeSys '19), pages 24-29, March 25, 2019.

Contribution: *The author suggested the main idea, designed the technical solution, and conducted evaluations. Dr. Nitinder Mohan took part in formulating the problem along with development of the solution's algorithm, workflow, and notably improved the publication's writing. The collection and preprocessing of data needed in the evaluation were performed by Dr. Walter Wong. The survey of the related work was done by Prof. Suzan Bayhan. Prof. Jussi Kangasharju gave his valuable insights concerning every aspect of the article and participated in the process of writing.*

Publication II: ICON: Intelligent Container Overlays. Aleksandr Zavodovski, Nitinder Mohan, Suzan Bayhan, Walter Wong, and Jussi Kangasharju. Published in ACM Workshop on Hot Topics in Networks (HotNets '18), pages 463-474, November 15-16, 2018.

Contribution: *The publication was led by the author, who offered the main concepts, addressed the technical aspects and the performance of the solution. Dr. Nitinder Mohan contributed to writing, technical and algorithmic design of the solution, reassuring interoperability, and coherence with the existing protocols and systems. Prof. Suzan Bayhan addressed the theoretical aspects of the solution's control mechanisms. Dr. Walter Wong gathered the evaluation data and gave valuable insights. Prof. Jussi Kangasharju actively participated in all phases of the work.*

Publication III: DeCloud: Truthful Decentralized Double Auction for Edge Clouds. Aleksandr Zavodovski, Suzan Bayhan, Nitinder Mohan, Pengyuan Zhou, Walter Wong, and Jussi Kangasharju. Published in IEEE 39th International Conference on Distributed Computing Systems (ICDCS '19). pages 2157–2167. Dallas, TX, USA, July 7–10, 2019.

Contribution: *The author proposed the concept of the publication, designed required protocols and algorithms, proved the properties of the presented auction mechanism, performed the evaluation of the solution.*
Prof. Suzan Bayhan co-supervised the work adding to every aspect of the publication: optimization problem, formal definitions, workflow and algorithms, writing, etc. Dr. Nitinder Mohan contributed significantly to evaluation data processing and its analysis, writing, and gave important comments. Dr. Pengyuan Zhou and Dr. Walter Wong provided valuable insights and assisted in writing. Prof. Jussi Kangasharju, the supervisor of the publication, contributed to writing, conceptual design, and other aspects of the work.

Publication IV: Anveshak: Placing Edge Servers In The Wild. Nitinder Mohan, Aleksandr Zavodovski, Pengyuan Zhou, and Jussi Kangasharju. Published in Proceedings of the ACM Workshop on Mobile Edge Communications (MECOMM '18). Pages 7-12. Budapest, Hungary, August 20, 2018.

Contribution: *Dr. Nitinder Mohan guided the work, formulating the problem, devising the methodology along with the solution algorithm and its implementation. The author, together with Dr. Pengyuan Zhou, performed data analysis and assisted in solution implementation. The supervision of the publication was conducted by Prof. Jussi Kangasharju, who provided valuable insights and contributed to writing along with the others.*

Publication V: Open Infrastructure for Edge: A Distributed Ledger Outlook. Aleksandr Zavodovski, Nitinder Mohan, Walter Wong and Jussi Kangasharju. Published in 2nd {USENIX} Workshop on Hot Topics in Edge Computing (HotEdge '19). Renton, WA, USA, July 9, 2019.

Contribution: *The publication was led and majorly written by the author, who suggested the main concept, conducted the analysis, and made the central insights. Dr. Nitinder Mohan and Dr. Walter Wang acquired the relevant data and presented it visually. Prof. Jussi Kangasharju supervised the work and contributed to writing.*

Manuscript I: Provisioning Services at Edge with Intelligent Containers. Aleksandr Zavodovski, Suzan Bayhan, Nitinder Mohan, Lorenzo Corneo,

Pengyuan Zhou, Walter Wong, Jussi Kangasharju. Under submission and review to IEEE Global Communications Conference (Globecom). Taipei, Taiwan, from 7 to 11 December 2020.

Contribution: *The manuscript is a continuation of work on ICON particularly focusing on a distributed decision-making algorithm. The author held primary responsibility for designing the algorithmic solution, evaluating it, along with producing the text of the work. Prof. Suzan Bayhan contributed significantly to the development of the solution and participated in writing. Nitinder Mohan performed analysis and representation of evaluation data, and also assisted in writing. Lorenzo Corneo aided remarkably in the development of the algorithm. Dr. Pengyuan Zhou contributed in writing. Dr. Walter Wong procured the evaluation data. Prof. Jussi Kangasharju, supervising the work, gave valuable insights and contributed to writing.*

1.4 Thesis Organization

The thesis is organized as follows. Chapter 2 concentrates on the state of the art in edge computing, emphasizing the technologies relevant for OpenIE. In particular, we examine edge application requirements and upcoming solutions aimed to address them, discuss the role of DLT in edge and cloud computing, and assess the potential of crowdsourcing. In Chapter 3, we describe our vision of OpenIE. Each section of the chapter is covering a particular component or problem of OpenIE: the discovery procedure of Elastic Extensible Cloud (ExEC), service overlay of ICONs, placement strategy of Anveshak, DeCloud distributed auction and the role of DLT in OpenIE. We conclude this thesis in Chapter 4, summarizing our work and outlining future perspectives.

Chapter 2

Edge Computing: State of the Art

This chapter elaborates on what has been accomplished by researchers and industry in the field of edge computing. The information presented here serves as an introductory background for the motivation and principles behind OpenIE. Since the field is vast and far-reaching, the focus will be on the most relevant topics considering our research questions, such as edge computing architecture, infrastructure, and platform solutions. We examine the requirement of various kinds of edge computing applications. Details of the initiatives for physical infrastructure deployment will also be covered along with crowdsourcing approaches. Efforts to develop market models for edge resource provisioning will deserve special attention. We will also cover how DLT are utilized in various aspects of edge computing.

2.1 Applications, Solutions, and Platforms

By its nature, edge computing is a multi-level setting consisting of clients, edge, and the cloud, as Figure 2.1 illustrates. With hierarchical approaches [60, 149], the edge layer is not necessarily flat but can be represented by a topology of computational and network devices. The network distance between cloud and edge is, in most cases, long, with the latency on the magnitude of hundreds of milliseconds, as Li et al. [119] suggest. However, as clouds tend to become more geographically spread, in practice, one might expect lower latencies since Li et al. [119] conducted their study in 2010. The entire purpose of the edge layer is to be close to its clients. Thus, depending on the nature of the application, the requirements for the delay may vary from tenths to a few milliseconds. Next, we describe edge applications along with the latency demands they exhibit.

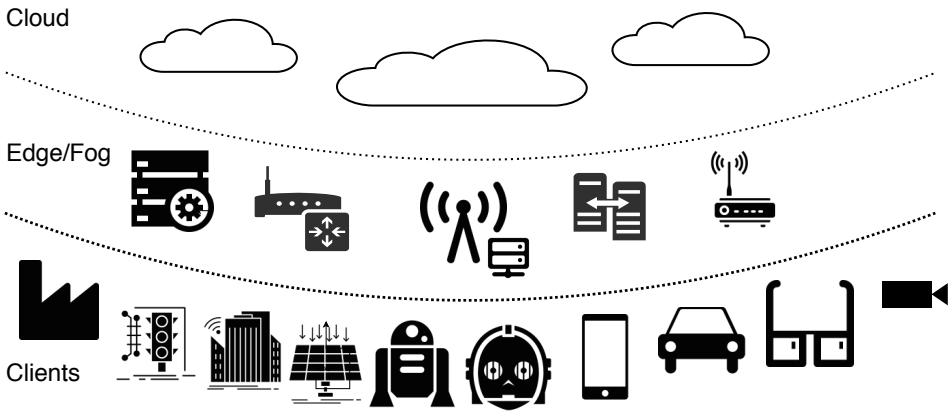

Figure 2.1: The three layers of edge computing.

2.1.1 Latency Requirements

We can roughly identify four categories of applications by their latency requirements. The most demanding are AR/VR applications, for which an important metric is Motion-to-Photon (MTP) latency. MTP latency is the time needed for reflecting a user's motion on a display screen. To create an *immersive* experience, the MTP latency must be less than 20 ms [128]. Moreover, as a study by the National Aeronautics and Space Administration (NASA) suggests, advanced head-up displays require MTP latency of less than 2.5 ms [13]. Next, follow the applications capable of operating within Perceivable Latency (PL) bounds. PL is the boundary at which the lag between the user's input and its effect on the displayed image becomes detectable by human perception [161]. The typical application is gaming, for which PL of 40 ms and below guarantees an excellent performance, while 100 ms is the boundary when input lag starts to be irritating for the players. The last border of latency-critical applications is Human Reaction Time (HTR), i.e., the delay between the actual appearance of stimulus and human motoric response to it. The value for the average human is around 250 ms [197], and applications such as remote surgery must operate within that bound. A skillful surgeon can perform the procedures safely when the networking delay is around 200 ms, but with more remote experience, this boundary can be raised to 500 ms [111]. The least demanding for the latency needs are IoT applications since they utilize the edge for aggregating upstream data flows instead of reducing communication latency. The exceptions are IoT scenarios involving actuator or robotic control, and for these cases, latency requirements depend

on a particular scenario. The applications, their latency requirements, and potential infrastructural solutions are summarized in Table 2.1. The infrastructural solutions for edge computing are *loosely* coupled with these four categories of latency requirements that we have described above, and the next section provides a systematic overview of these solutions grouped by the categories of applications that they intend to serve.

2.1.2 Infrastructure Solutions for Edge Computing

Here we examine main trends in physical and virtual infrastructure development of edge computing, concentrating on their capabilities to serve applications with various latency requirements, which we grouped into the Categories I-IV in Table 2.1. It is worth noting, although we discuss, e.g., cloudlets and MEC as separate approaches, the division is rather tentative since one can see MEC installations as cloudlets connected directly to telecommunication masts. Moreover, despite the fact that we consider MEC to be the fastest edge for mobile clients, alternative scenarios are also possible, e.g., nearby cloudlets with WiFi connection may provide a better Quality-of-Experience (QoE) to the client. Thus, division by latency groups is also tentative. However, we find the loose taxonomy presented in this section convenient since it is based on the entities backing the development of particular approaches, giving them unique traits and consistent paths for the future.

Multi-access Edge Computing (MEC)

Developed by ETSI in symbiosis with MNOs, MEC [58, 182] is probably the most ambitious undertaking among other plans for edge computing infrastructure deployments at the moment. Coupled with the future 5G cellular network, MEC aims to satisfy the most stringent requirements of Category I applications, listed in Table 2.1. Since the consortia of MNOs back MEC, servers can be placed directly within telecommunications towers or in the nearest possible proximity to them. Ideally, the edge server will be on the next network hop from a device using a cellular connection. Given the next generation of cellular networking technology, namely, 5G, the latency on that hop (so-called "air latency") is estimated to be from below five up to 12 milliseconds [82,195]. A recent measurement study performed by Narayanan et al. [143] on a commercial 5G network suggests 27 milliseconds. The subject has not yet been thoroughly researched, as more empirical data is needed to draw final conclusions. However, in the case 5G will redeem its promise, MEC will be capable of satisfying the majority

Category	Latency Group	Latency, ms		Application Examples	Infrastructural Solutions
		Boundary	Optimal		
I	Motion-to-Photon (MTP)	< 20	≈ 2.5	AR/VR, head-up displays	5G MEC
II	Perceivable Latency (PL)	< 100	≈ 40	Gaming, Video streaming	Cloudlets,
III	Human Reaction Time (HTR)	< 250	≈ 200	Remote surgery, Teleoperated machinery	Edge Clouds.
IV	Not critical for upstream aggregation, may vary for actuators or robotic control.			Miscellaneous IoT: smart cities and homes, electrical grids, manufacturing, agriculture, warehousing, etc. Driving assistance, offloading, robotic control, drones, etc.	Fog, Mist Clouds, etc. Ad-hoc edge clouds, mobile and vehicular clouds, crowdsourcing, etc.

Table 2.1: Edge applications, latency requirements, and infrastructural solutions.

of Category I applications, probably except for the most challenging ones. For the mobile handheld devices with a cellular connection, MEC will likely to be the closest edge infrastructure provisioned in an organized fashion by large commercial entities. The reference architecture for MEC [57] provides a clearly defined set of components and interfaces for their interaction. The platform hosts mobile edge applications in a virtualized environment, enables discovery of those applications by clients, and configures local DNS services to route requests appropriately. The key enabling technologies for MEC are Network Function Virtualization (NFV), Software-Defined Networking (SDN), and hardware virtualization, with support for both VMs and containers. Joint optimization of NFV and MEC services is possible, and there are efforts to make MEC capable of operating in existing NFV environments. Also, some of NFV components can be at least partiality utilized by MEC, such as Management and Network Orchestration (MANO). MEC provides convenience for the implementation of mobile service migration scenarios. As for any other distributed edge platform, the following challenges of service orchestration are also relevant for MEC: i) resource allocation ii) service placement iii) edge selection iv) reliability insurance. While MEC can rely on NFV and other orchestrators in some of the above issues, particularly problems of service placement and edge selection might need additional investigation in the future.

2.1.3 Cloudlets and Edge Clouds

Suggested by Satyanarayanan et al. [167] in 2009, the cloudlets spawned the inception of edge computing in its present form. Cloudlets can be seen as a logical continuation of the Mobile Cloud Computing (MCC) trend, which emerged in the first decade of the millennium soon after the establishment of the cloud computing paradigm [47]. Although originally intended as a "data center in a box", which is connected to its clients by a one-hop WiFi link, at present, it is rather difficult to see massive deployments of cloudlets following that vision. In this thesis, we augment the cloudlet concept by adding edge computing datacenters that CSPs are deploying or plan to deploy, so that a cloudlet can be mini- or even mid-sized datacenter instead of "box", i.e., a small cloud, as the name implies. Therefore, most likely, cloudlets will not be located on the first network hop from the clients and will be outperformed by MEC in terms of latency, at least for mobile handheld devices using cellular connections. Thus, we expect them to serve applications of Categories II and III (see Table 2.1).

As the initial concept [167] suggested, the cloudlets were supposed to utilize standard cloud technology to ease interoperability with the cloud.

Indeed, industry, academia, and community are largely favoring this approach, and today many cloud platforms and technologies have their own adaptations or extensions for edge environments. This includes OpenStack++ [54], a flavor of OpenStack [151] developed by Satyanarayanan's team specifically for cloudlets, KubeEdge [145] – an edge version of the widely used Kubernetes [146] cluster manager, DC/OS [43, 205] has extended its operation to the edge, and the list may be continued. The virtual infrastructure backing cloudlets is quite advanced at the moment, as there are industry-grade components and platforms covering almost every aspect of cloudlet operation. Moreover, there is the Open Edge Computing Initiative (OEC), formed by the alliance of Carnegie Mellon University with major companies in the IT industry.[1] The goals of OEC are to promote *convergence* of edge technologies and platforms globally, arrange demonstrations showcasing the edge capabilities, establish test environments, increase industrial participation, address the research challenges, etc. Despite the activities of OEC not yet being very intensive, and it is uncertain how industrial participants will incorporate suggested ideas, the initiative is at the beginning of its journey.

When it comes to physical infrastructure, we have not yet witnessed notable deployments of cloudlets on any scale of significance. Nevertheless, CSPs have remarkably increased their presence in various geographical regions and plan for more, as can be seen in, e.g., the Microsoft Azure coverage map [134]. Those numerous cloud facilities displayed on the Azure map are convenient cloud datacenters by their nature and do not display any special technical or administrative cloudlet-like features. However, more initiatives are upcoming from the CSPs, which are likely to have a cloudlet flavor. Google is seeking an alliance with MNOs to establish a general-purpose infrastructure that will compete with MEC. In contrast to MEC, the solution will rely on proprietary technology, Kubernetes-based Anthos [117], instead of standards developed for MEC by ETSI. AWS has similar ambitions, seeking to provide low latency service in densely populated areas in cooperation with MNOs [49]. In case CSPs initiatives will succeed, we might witness the practical realization of the cloudlet vision. However, this scenario is likely to differ remarkably from the original concept in its reliance on proprietary technologies and participation limited to large commercial players.

[1] At the moment of writing, among the participants were Intel, Microsoft, Nokia, NTT, Seagate, VMWare, Vodafone, etc.

2.1.4 Internet-of-Things and Fog Computing

In the context of IoT, edge computing is generally referred to as fog. Up till now, this is an application area where probably most of the practical advancements have happened, as we will see in this section. In many cases, IoT applications are not very sensitive to delay, as the primary motivation is to aggregate and preprocess, e.g., sensor-generated data locally, avoiding an excessive traffic generation and storage space requirements that would occur in the case of sending all of the raw data to the cloud. However, in certain application areas, e.g., involving actuators control or robotics, delay tolerance may be quite low. IoT scenarios are grouped under Category IV in Table 2.1, and this category stays apart from the rest as it does not assume, in general, existence or establishment of general-purpose edge computing physical infrastructure. The entities in possession of IoT environments are the ones expected to take responsibility for the economic and administrative overhead caused by the required fog servers deployment. The opposite is possible, i.e., IoT installations could potentially utilize generally available edge servers. However, this opportunity can be limited, e.g., by the technologies intrinsic for sensor networks and the need to place computing nodes exactly at the specific locations of the network topology. For instance, general-purpose machines are not likely to support power-efficient communications enabled by the Long Range (LoRa) Low-Power Wide-Area Network (LPWAN) protocol.

Generally, major CSPs are offering their solutions for IoT in the form of a computational appliance that customers can acquire and deploy at their premises. The device, supporting an IoT technological stack, is manageable in similar fashion along with all the other services that CSP offers and integrates seamlessly with the applications installed in the cloud.

Such solutions are Azure Stack [135] and AWS Outposts [11], which in essence, provide the means to establish a hybrid cloud environment. This approach extends the capabilities of conventional cloud datacenters with a possibility to place, e.g., data preprocessing, IoT machine learning, and analytics, along with other delay-sensitive services at customers' premises.

The offerings of IoT computational devices are not limited to CSPs, and many hardware manufacturers are aiming to fill the new market niche. As the scope of this thesis is limited, we mention Cisco as a pioneer of fog computing [21], which covers almost all aspects of setting up an industrial IoT infrastructure environment [36], including routers [37] and compute nodes [38].

Smooth integration offered by CSPs and vast hardware availability are not the only strengths of fog computing, namely, there is the Industrial

Internet Consortium (IIC) [102] of key industry players and academia[2] backing the development of common standards and promoting the reference fog architecture [149]. Compared to OEC, IIC is at a more mature stage of development, as the reference fog architecture was standardized by the Institute of Electrical and Electronics Engineers (IEEE) in 2018 [96].

An ultimate form of fog is *mist* computing [166], where IoT sensors perform limited computational tasks by themselves. Conceptually close to mist is *dew* computing [178], which also extends the cloud metaphor further and assumes usage of client devices, e.g., smartphones or notebooks as compute nodes.

2.2 Distributed Ledger Technologies in Edge and Cloud Computing

Despite being relatively new, DLT have already established themselves in many diverse areas of application, and edge computing is no exception. We start by a concise description of DLT along with underlying operation principles and then proceed to depict the role of DLT in edge and cloud computing.

The most well-known examples of DLT make use of a blockchain, which was largely popularized by Bitcoin [142]. However, it is worth to note, that blockchain is only a data structure used to store information, and there are other distributed ledgers harnessing various decentralized consensus protocols but not utilizing blockchain for secure storage, e.g., Ripple[3] or IOTA[4]. We use DLT as an umbrella term, referring to fundamental features of distributed ledgers, such as decentralized agreement and tamper-proof storage. Since the description of the technological abundance of the DLT field is out the scope of this thesis, we limit our background information to blockchain-based platforms. A blockchain is an ordered sequence of blocks, hence the name. Blocks are records, and each one of them encompasses a reference to the previous block; Proof-of-Work (PoW)[5] hash used to cryptographically protect the content of the block; timestamp; useful payload, which may be a set of financial transactions or any other information, depending on the purpose of the system. Blocks are generated by the

[2]Currently, members of consortium include among others Princeton and Shanghai Universities, AT&T, Cisco, Dell, Intel, Microsoft, GE, ZTE, and IEEE.

[3]https://ripple.com/

[4]https://www.iota.org/

[5]PoW is not the only way to protect the block, and as we see later, there are alternative, more sustainable choices.

Peer-to-Peer (P2P) network of participants according to the shared protocol standard. Those peers, called miners, exchange the generated blocks with each other, and cross-validate them. Thus, each miner stores its own copy of the blockchain, which contains all the events of the system. There are two kinds of blockchain networks: *permissionless* or *public*, where anyone can join as a miner, and *permissioned*, in which joining the network required additional approval from other miners or some third-party authority. Generally, the security of the DLT system may be breached only if the majority of the miners collude. Therefore, Byzantine fault tolerance of DLT platforms is estimated to be quite high [165].

Another milestone for DLT was achieved with the launch of the Ethereum [67], the first platform to implement smart contracts. Suggested by Nick Szabo, smart contracts are "building blocks for digital economy" [181]. In a DLT with smart contracts, miners have an additional function: to execute code and collectively verify the execution. Essentially, a smart contract is a programs, that can be invoked by anyone sending a contract call message to the P2P network of miners. A miner will receive the message, execute the code of the contract, and include the results in the next block. Other miners will receive the block and verify the correctness of execution, upon success accepting the block as a valid one. These features of smart contracts make them attractive as a means for reaching multi-party agreement without a centralized authority. Next, we describe the role of DLT in edge computing.

Remarkable attention towards the distributed agreement and security that DLT can offer is originating from the field of IoT [61]. Logically, considering the IoT environment on the scale of a smart city, the need for decentralization is well-motivated, as myriads of devices belonging to different administrative domains need to interact seamlessly while keeping reliable accounting records [76]. Rahman et al. [162] utilize the blockchain platform as a distributed storage for data collected by edge and fog nodes to facilitate the shared economy services of a smart city. Zhao et al. [212] use blockchain distributed storage and security features for crowdsourced federation learning of IoT usage patterns. Distributed fog-cloud architecture empowered by blockchain is suggested in [174]. Khan et al. join blockchain and edge computing together in their architecture for participatory smart city applications [114]. Data exchange for smart toys built on top of Hyperledger Fabric [69] is devised by Yang et al. [202].

While in IoT distributer ledgers are used mainly for secure data storage with, e.g., record origination validation capability, authentication, etc., there are attempts to use distributed ledgers in a way that is more rel-

evant from the OpenIE perspective. Namely, utilizing DLT as a control fabric, decentralized transaction processing system, dynamic SLA negotiation framework, digital agreement, and many others. For instance, smart contracts are used in prototyping of a next-generation cloud [109,112,147]. Smart contracts are handling tenant management and the use case is likewise applicable to edge computing. Wright et al. develop a smart contract specifically for edge purposes [198] with an escrow capability. In case the edge node would not provide the service at the level of agreement, the escrowed funds would be returned to the mistreated client. A control system for the edge computing complying with IEC 61499 (standard for distributed control systems platform) is developed in [180]. The system utilizes Hyperledger Fabric [69] technology. SLA negotiation and enforcement with DLT is explored in [215]. The authors enhance the Ethereum platform with a special witness role to achieve the wanted level of trustworthiness. The utilization of blockchain in SLA handling is investigated from various angles in [18,170,171,173,188,214], reaching beyond the scope of cloud and edge computing.

The use cases and examples above are just a small fraction of the ongoing work in the area where DLT and edge computing cross, and much more information is systematically summarized in [19,61,62,72,76,113,184,203]. In our work [210], which we discuss in the next chapter, we examine the strengths and shortcomings of state-of-the-art distributed ledgers as enablers for OpenIE.

2.3 The Potential of Crowdsourcing

Disruptive innovations in the field of DLT have made a positive impact on the crowdsourcing scene in general. For instance, a highly recognized scientific computing platform BOINC [6] has started to reward contributors in Gridcoin cryptocurrency [83]. From the perspective of edge computing, crowdsourcing could be an attractive path for development since computational devices available at general public disposal are usually located at the edge of the network, exactly where they are needed the most. However, as we already noticed, so far, there has been no breakthrough comparable to BOINC and blockchain systems supported by ethically or economically motivated volunteers. Nevertheless, there are active efforts coming from both academia and industry sides seeking to unfold this hidden potential.

2.3.1 Research Endeavors

Kuendig et al. [116] take a systematic approach towards establishing the foundations of what they call a Crowdsourced Edge Computing (CEC) paradigm. In concordance with [93], the authors identify the following distinctive features of a well-established crowdsourced system: (i) diversity – a capability to support heterogeneous resources, both devices and infrastructure; (ii) independence – the ability of crowdsourced entities to operate autonomously (iii) decentralization – implied by the previous two, centralized governance mechanisms should be avoided whenever possible (iv) aggregation – the possibility to assemble the final result from individual peers' contributions. These criteria will be useful also for us in assessing other crowdsourcing proposals. Kuendig et al. define CEC at the high-level architectural description and identify research challenges, such as ad-hoc network formation and distributed routing, general-purpose computation, resource management, QoS instability because of unreliable nodes, and others. Despite the severity of those, the authors consider CEC to be feasible and beneficial. The work of Hosseini et al. [93] is ambitious in aiming for the creation of the crowdsourced equivalent of the cloud. The methodology the authors use is applicable to edge computing as well. However, the presented implementation is at the level of proof-of-concept and relatively far from being ready for practical engagement by the public.

While the above concentrates on the outline of a technical framework, however, the paramount issue of crowdsourcing is the incentivization of the peers, i.e., how to make participation in the system attractive. This is closely related to game theory and economics, and there are numerous works on the topic. Jiang et al. [108] model the setting of mobile edge caching as a two-stage Stackelberg game, improving the welfare of participants significantly. Despite the scenario being rather CDN-flavored, the main ideas might be utilized in common applications of edge computing as well, including IoT. Xu et al. [201] present an *incentive mechanism* intended to empower the crowdsourced market for the edge-cloud environment. The framework features a double auction with a welfare maximization objective. The work remains on a highly theoretical level, yet leaving some questions critical for practical engagement unclarified. In our proposition – DeCloud [206], we concentrate on a similar topic and present the Dominant Strategy Incentive Compatible (DSIC) double auction mechanism for edge-cloud resources provisioning. We take a different theoretical approach than [201] and sketch a clear pathway to our framework implementation on a blockchain platform. The next chapter contains a comprehensive description of DeCloud's role as a building block of OpenIE.

2.3.2 Existing Systems

Dfinity [187] promotes itself as "the Internet Computer", targeting to offer a serverless-like infrastructure for running applications on a worldwide scale. Potentially any datacenter can be registered with Dfinity and benefit from the additional load. Dfinity features custom technical solutions, such as a decentralized protocol called Internet Computer Protocol, tailored network abstraction – Network Nervous System, customized programming language Motoko, and an advanced blockchain platform. Although Dfinity is positioning itself as an alternative to monopolistic CSPs, it could also function as an edge service provider in case there are datacenters in close proximity to the clients. Unfortunately, the information of existing datacenters, which are registered with Dfinity, is not currently available.

Targeting edge computing directly, Edge Network [51] shares many similar concepts with Dfinity. However, the topology of the network is public information, and at the moment of writing, there were 659 hosts [52]. Any Linux device capable of running Docker can join the system. Interestingly, the procedure of becoming a member of the platform requires an initial investment of 5000 $EDGE tokens, which can be obtained at major cryptocurrency exchanges and, at the moment of writing, were exchanged at the price of $0.015746 USD, making the total value of initial stake $\approx$ 78 USD. The investment can be seen as risky since, during the launch in 2018, the price of the token was fluctuating around 0.4 USD. The reward for the peers comes in the form of Proof-of-Stake (PoS) protocol minting, which in practice means that the initial stake of 5000 tokens will generate $\approx 15\%$ gains annually, according to [51].

Sonm [179] is another example of a blockchain-empowered decentralized computation network, addressing Fog, CDN, and Machine Learning (ML) needs. Contrary to Edge Network, no initial stake is required and peers are immediately rewarded for the computation they perform. The number of active nodes is 12 at the moment of writing. The platform utilizes Docker containers as a way to package tasks.

Golem [79] and iExec [100] seek mainly to provide an alternative to cloud computation, with the technology stack alike to previously discussed systems, meaning that certain edge computing scenarios would also be technically feasible for them. Golem promises to run any code compiled to WebAssembly (WASM) [192], which gives a fair selection of programming languages to create tasks for the platform. The platforms are not massive in scale, e.g., Golem's participants supply 961 CPU cores and 1.75 TiB of RAM in total at the moment of writing [78], while iExec does not provide this information.

Generally speaking, it appears that the enthusiasm for such platforms, initially ignited by the success of Ethereum launch in 2015, is somehow decaying. E.g., in June 2018, there were 3000 CPU cores in the Golem network [156]. However, it might be too early to draw any conclusions as not much time has passed since DLT became widely recognized, and the development in the field continues to be innovative and hectically paced.

2.3.3 Summary: The Missing Parts

To summarize, there are many solutions focused on economics, which depending on their goal optimize for, e.g., (near) maximum welfare, revenue for the providers, minimizing costs for clients, fairness, and so on. While the theoretical findings of these works in most cases can be incorporated into existing or upcoming crowdsourced systems, as we previously concluded, none of these platforms has reached the crowds at any remarkable scale yet. With the given abundance of theoretical work on economic incentives and DLT platforms making handling at least of cryptocurrency transactions easy, we have a toolbox of unprecedented scope for creating crowdsourced systems.

However, in contrast to the upcoming systems attempting to attract participants by some sort of fiscal stimulus, BOINC became popular as a platform for the SETI@home[6] project, driven not by monetary incentives but the fascination for the search of extraterrestrial life. Likewise, during its early stages, Bitcoin [142] was supported by ethically or ideologically motivated miners craving a transparent currency that is not controlled by any central bank and, therefore, not subject to inflation. What is currently missing in the case of edge computing are exactly those immaterial values that made blockchain and crowdsourced scientific computing so successful.

Fortunately, there are reasons to support edge computing, having not only profit in mind. Sustainable development and edge are not very often mentioned together, but we hope that can be the potential source for enthusiasm. Namely, increasing the utilization level of devices belonging to the general public instead of establishing new small datacenters might be a reasonable environmental choice, especially in the case when such devices are running on the excesses of locally produced renewable energy. Green computing is not the only encouragement. The opposition to "feudalism" (as Liu et al. [121] characterize the current state of the Internet), and the monopoly of big companies can likewise be an effective driver of crowdsourcing. The role of a *digital equality* enabler that edge computing can uptake

[6] https://setiathome.berkeley.edu/

in regions where CSPs have limited coverage might also spark enthusiasm for participation in crowdsourced edge physical infrastructure.

The topic of crowd engagement and retention in the context of crowdsourced edge computing is out of the scope of this thesis. Still, it poses an interesting multidisciplinary research opportunity in the cross-section of computer science, economics, sociology, and psychology. Currently, crowdsourcing as a means of edge resource provisioning is facing complex challenges of technical deficiencies, security issues, and unclear motivation for the involvement of potential participants.

Chapter 3

Open Infrastructure for Edge Computing

This chapter outlines the principles and main concepts behind OpenIE and covers our contribution to the subject. We describe our vision of a global-scale edge discovery mechanism besides the operation of ICON service overlay and evaluation of its performance. We present our novel edge placement strategy that, in addition to the distribution of potential users, takes into account also the availability of edge resources. Further, we concentrate on the DeCloud framework featuring truthful double action adapted specifically for decentralized execution on DLT platforms. Having defined the market model for edge resources provisioning, we examine the potential of DLT in the context of OpenIE, and also for edge computing crowdsourcing solutions in general. Thus we address most of the major challenges that OpenIE poses, which are discovery, autonomous dynamic governance, economic model, and coordination along with transaction handling by a decentralized authority.

3.1 Overview of OpenIE

The motivation for OpenIE is to make edge computing pervasive. To be capable of serving most potential applications and clients, edge server deployments need to cover vast geographical areas. Such a dispersed nature of computing capacity installations requires, on the one hand, interoperability between various edge platforms so that services can without let or hindrance migrate as close as possible to their clients across administrative borders of various systems. On the other, the threshold of becoming an edge service provider must be lowered to its extremes. Addressing both

issues mentioned above, and, moreover, providing an economic incentivization for providers, OpenIE can catalyze an emergence of the edge service provisioning market. On the supply side of such a market, there will be Independent Edge Providers (IEPs) – entities conforming to OpenIE practices that wish to offer their computing capacity for the needs of edge computing. Organizations in possession of local datacenters, individuals who installed appropriate crowdsourcing platforms with OpenIE support on their smartphones or PCs, major CSPs, or MNOs in ownership of cloudlets or MEC servers – any of these can become an IEP. The demand side will be formed by the application owners developing edge applications for various purposes.

OpenIE assumes that application development will follow the practices of microservices architecture, i.e., the application will consist of services that are packaged in virtualized entities, such as containers, VMs, or even unikernels. Among those services, some will be intended for placement at the edge. The others will be performing better in the cloud. Virtualization technologies can secure IEPs from malicious clients, while TEEs can be employed for client protection from non-trusted IEPs. In its functioning, OpenIE relies only on existing and already deployed networking protocols and technologies, requiring no modifications or enhancements to the present infrastructure. Logically, the operation of OpenIE starts with the discovery process – when the application autonomously[1] detects the available IEPs to place its edge services near the clients. Next, we elaborate on the principles of the discovery process.

3.2 Discovery of Independent Edge Providers

The discovery process is an integral part of ExEC – a framework facilitating IEPs and dynamic service onloading[2] to the edge. ExEC [209] is attached to this thesis as Publication I. Figure 3.1 illustrates the operation of ExEC. Initially, the entire application is deployed in the cloud, and *ExEC Orchestrator*, the main component of the solution, starts to monitor incoming requests. From Internet Protocol (IP) addresses of the clients, the orchestrator infers subnets, from which the requests originate. If the subnet is previously unseen, the system launches the discovery procedure.

[1] Autonomous discovery is the most advanced scenario needed for highly dynamic applications with spatiotemporally varying load. In the cases where an application can perform well with static deployment of edge services, IEPs registered according to OpenIE practices can be located during initial deployment of the application.

[2] In contrast to offloading where the client sends computational tasks to an edge server, onloading is the process where service initially residing in the cloud is moved closer to its client, i.e., to the edge.

In the optimistic case, when there are IEPs on a path to the subnet (or in the subnet itself) where new clients reside, the orchestrator negotiates with the IEP (e.g., if there is any capacity available, what the price is, etc.) and onloads the service to the newly discovered deployment location. The clients are redirected to use the service onloaded to IEP instead of the one deployed in the cloud.

Figure 3.2 depicts the discovery procedure. The first step that the orchestrator carries out is the identification of the path to a new client. There are various technical ways to accomplish this task, but the most straightforward and generally available method to perform basic network tomography is the traceroute tool [127]. Traceroute returns not only on-path IP addresses but also latencies between them. From traceroute, the ExEC orchestrator receives the list of IP addresses constituting the path. The second step is to find out domains registered in the Domain Name System (DNS) on the path. This is done by issuing reverse DNS lookups [28, 97, 98] on the IP addresses obtained in the first step. Reverse DNS lookup returns the domain name for the IP address. Having a list of on-path domain names (in our example, these are domains A and B, see Figure 3.2), the orchestrator proceeds to the third and the final stage of the process. During this phase, the orchestrator performs DNS SRV queries [99] to on-path domains for *edge* records.

Given that IEPs have added SRV edge records for their edge services in the DNS, they will be discovered by ExEC. The primary assumption of the discovery mechanism is the existence of edge SRV records in DNS. Positively, such an approach enables a global-scale solution without any modifications to the existing networking infrastructure or protocols. The downside is that IEPs will have the administrative burden of adding records to DNS. However, the overhead is rather minor compared to the ability to be easily located by the customers generating the revenue.

While the approach that ExEC employs is onloading, i.e., migrating the service from the cloud closer to its client, there are no impediments for other scenarios. For instance, in the offloading case, mobile devices can use the discovery procedure by querying the domains on the first hops of their network path for the availability of edge servers.

The discovery does not have to be limited to the on-path technique we just described. Using the vast network topologies maintained by, e.g., the Center for Applied Internet Data Analysis (CAIDA)[3], ExEC, or any other entity motivated in discovering edge facilities can identify domains, which are in close networking proximity to their clients (or themselves) and query

[3]https://www.caida.org

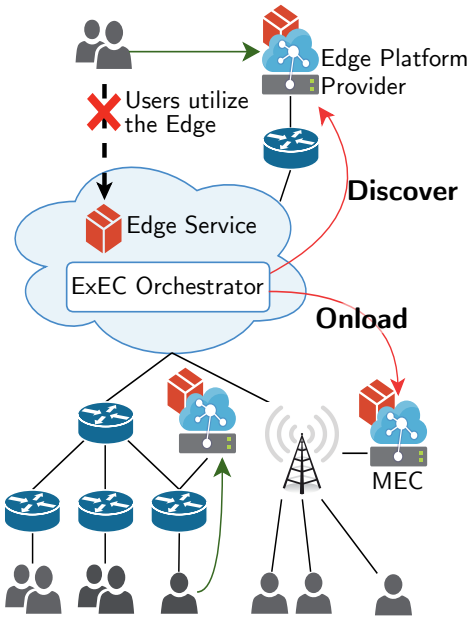

Figure 3.1: Overview of ExEC operation.

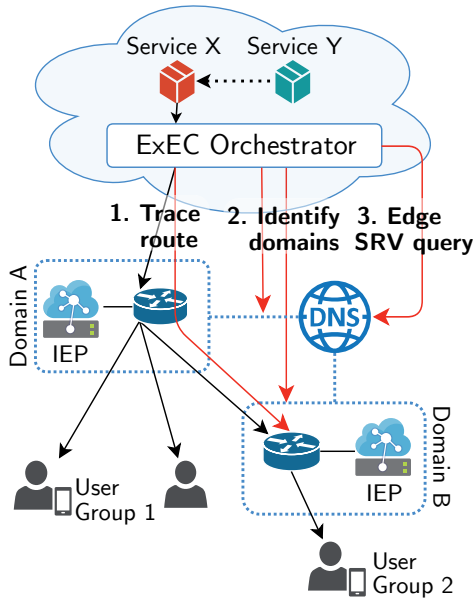

Figure 3.2: Discovery of IEP in ExEC.

them for IEPs. This kind of approach would not require using traceroute or its equivalents, and the only remaining prerequisite is the registration of IEPs in the DNS.

3.2.1 Negotiation with IEPs and Container Yard

Before deploying a service to IEP, the ExEC orchestrator needs to reach an agreement with IEP on the details of the deployment. On the IEP side, the *container yard* application will handle the process. We suggested container yard in [208] (which we will describe in Section 3.3), and it was not originally part of ExEC.[4] However, it is beneficial that both solutions share the same container yard platform, which serves as an intermediary between them and under the hood technology that IEPs employ, such as cluster management systems and schedulers, e.g., Docker Swarm [48], Kubernetes [145, 146], DC/OS [43], and others. Container yard exposes a RESTful service to the outside world, which entry point (contained in the edge SRV record) the orchestrator discovers as a result of the procedures described in the previous section.

[4] In [209], IEPs were supposed to run ExEC management service, basically equivalent in its functionality and role to container yard.

To initiate a negotiation, the ExEC orchestrator sends a *hello* message to the container yard of IEP. Some preferences might be included in the message, such as the time period for the service deployment. The reply message contains the list of available time slots for deployment, hardware resources and networking bandwidth available during these time slots, prices, preferred ways of the contractual agreement and financial transaction handling. In cloud computing, the latter two are generally handled by some third-party broker, e.g., [42, 92]. However, more agility to OpenIE would bring harnessing of DLT, namely, the smart contracts. While Section 2.2 provides preliminary knowledge on DLT, and Section 3.6 covers the topic of DLT in OpenIE, here we briefly overview the process for the smart contract case.[5] In systems like Ethereum [196], smart contracts have unique addresses, and the address of the contract that IEP uses must be included in the reply message sent back to the orchestrator. The contract encapsulates details of SLA (or reference to it), pricing, and other appropriate information. Whether the orchestrator finds the offer by IEP acceptable, it invokes a preliminary agreement method of the smart contract. Parameters for the method invocation include the information on the time slot that was chosen, preferred hardware configuration, etc. Whether the time slot is still available, the IEP calls the method of smart contract that confirms the preliminary agreement. Otherwise, it invokes the rejection method. As a result of confirmation, the smart contract takes the amount of digital asset-bearers or tokens[6] that deployment costs into escrow and notifies the counterparties, i.e., the orchestrator and IEP. After the IEP has provisioned the service as agreed, the orchestrator closes the deal by calling the finalization method of smart contact. The call results in the transfer of the escrowed tokens to the IEP account. As DLT and digital agreement, in general, are relatively novel approaches, there are some challenges in the process that we described above. Thus, we elaborate more on the topic in Section 3.6.

3.2.2 Service Placement and Evaluation

Once the IEPs are discovered and the orchestrator is aware of current topology and client flows, it needs to decide on which IEPs to deploy services. Logically, the best possible QoE would be achieved by deploying services on

[5] ExEC does not require smart contracts or any other DLT to function. The contact information may be returned to the orchestrator as a free text, in which case the deployment of the service must be done by the administrative personnel manually. However, it would be highly inconvenient.

[6] In Ethereum, these are represented by Ether cryptocurrency.

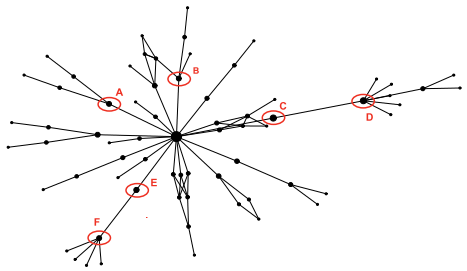

Figure 3.3: Fragment of the network topology. Nodes with highest betweenness centrality are circled red (except the central one).

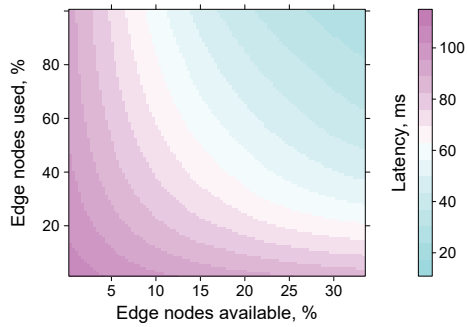

Figure 3.4: Centrality placement performance.

all IEPs, that are closest to the clients. However, in reality, the operation of the system will be restricted by budget limitations, so the orchestrator will have to choose a subset of IEPs. In ExEC, we have evaluated the placement strategy based on betweenness centrality [71]. Figure 3.3 gives an intuition of the concept, nodes with the highest betweenness centrality are circled in red except for the central one, which represents the cloud where the orchestrator is placed. We evaluated the performance of betweenness centrality using the topology constructed from the CAIDA public router dataset [95], limited to the East Coast of the US. After grouping the nodes with a subnet mask of 16 bits, we obtained a tree of 240 nodes. Figure 3.4 displays the performance of the system in terms of the average latency achieved by using a certain fraction (y-axis) of the edge nodes available in topology (x-axis). The substantial reduction in latency is achieved when the fraction of nodes with edge capabilities in the topology exceeds 20%. Publication I contains more technical details and evaluation results.

3.3 Intelligent Container Overlays

ExEC is a centralized system governed entirely by the orchestrator. Such a solution can perform well for limited scenarios. However, on a larger scale, the downsides of the centralized solution, such as low scalability, slow reaction time, excessive communication between the orchestrator and IEPs, become more evident. Thus, we propose Intelligent Container (ICON) – an entity encapsulating a service, which is aware of its environment and capable of autonomous decision making. ICON [208] is attached to this thesis as Publication II.

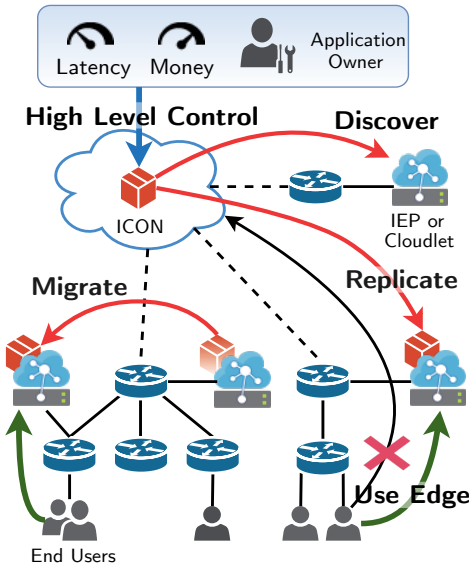

Figure 3.5: Overview of ICON operation.

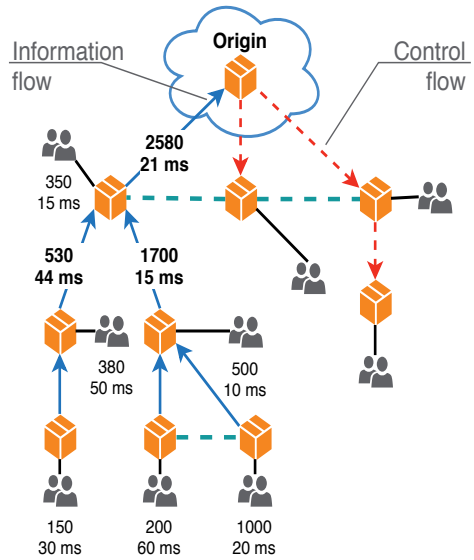

Figure 3.6: Overlay tree of ICONs.

Figure 3.5 gives an overview of an ICON operation. Similarly to ExEC, ICON monitors incoming requests and makes use of the ExEC discovery procedure upon identifying new requests arriving from previously unknown subnets. Once IEPs are located, ICON may choose to migrate to one of those newly discovered locations or deploy replicas of itself on chosen IEPs. Alternatively, ICON can decide to terminate itself in the case of low utility, delegating its remaining clients to upper-level containers. Each new replica will be a full-fledged ICON itself, performing the same tasks, and having autonomy in its decision. It will keep a reference to its parent, thus, the overlay of ICONs will eventually form a tree, as Figure 3.6 shows. The overlay structured as a tree provides an efficient way of communicating control and performance information. The latter is aggregated to reduce communication overhead, i.e., the child passes its parent the sum of all requests served and averaged delay information. ICON uses container yard application and the same negotiation procedure as we describe in Section 3.2.1 for ExEC. However, due to the distributed nature of the overlay, the container yard needs to perform additional functions related to synchronization, which we explain in the next section.

3.3.1 Control and Decision Making

The convenience of the application control is another challenge that ICON addresses. The overlay is governed by the utility function, simplifying the control activities of the application owner to adjusting the balance between price and latency performance.[7] Anytime the application owner changes priorities, new parameters are propagated from the root of the tree, called origin, down to the leaves, and used by all ICONs in the calculation of IEP utilities, which determine subsequent actions of ICONs. Formally, ICON $i \in I$ computes expected utility u that IEP $j \in J$ will achieve during the next deployment time slot $t+1 \in T$ as follows (omitting i and $t+1$ subscripts for clarity):

$$u(j) = w\left(\frac{\overline{r}_j}{\nu \bar{l}_j}\right) + (1-w)\left(\frac{\overline{r}_j}{c_j}\right), \tag{3.1}$$

where ν is a normalization factor to enable usage of single weight $0 \leq w \leq 1$ for balancing between expected average latency $\bar{l}_j > 0$ and cost $c_j > 0$; $\overline{r}_j$ is the expected number of requests. Values $\bar{l}_j$ and $\overline{r}_j$ expected during $t+1$ are obtained by computing an exponential moving average of previously observed l_j and r_j, respectively. The normalization factor ν is defined as follows:

$$\nu_{i,t+1} = \frac{\sum_{j \in J_{i,t+1}} c_j}{\sum_{j \in J_{i,t+1}} \bar{l}_j}, \tag{3.2}$$

where $J_{i,t+1}$ is a set of all available IEPs to ICON i for the next time slot of deployment $t+1$.

The utility of IEP (3.1) is a multi-objective problem since it depends on cost and latency. The solution to such a problem is a Pareto front, i.e., a set of Pareto optimal points [155]. For any of the Pareto optimal points, it is impossible to improve any of the objective functions without negatively affecting some other objective function(s). The common approach for solving multi-objective problems is the weighted sum method, and if all weights are positive (as in our case), the method results in Pareto optimal solution [130,131]. However, solving the multi-objective optimization problem might be too demanding for resource-limited ICONs. Thus, we present Algorithm 1 of complexity $\mathcal{O}(n^2)$ in the worst case.[8]

[7] The possibility for monitoring and manual control of individual ICONs is also technically feasible but not intended as a regular practice.

[8] Manuscript I contains the proof.

Algorithm 1 ICONs Decision-making

▷ Compute utilities of discovered IEPs
1: $I \leftarrow$ set of all deployed ICONs
2: $J_i \leftarrow$ discovered IEPs of ICON i
3: $\Omega \leftarrow$ IEP where ICON i is deployed (root of J_i)
4: $U_j \leftarrow \{u(j) : j \in J_i\}$
▷ Traverse tree of discovered IEPs bottom-up and combine IEPs to increase utility
5: levels $\leftarrow group(\{depth(j) : j \in J_i\})$
6: **for** level $\in$ levels **do**
7: **repeat**
8: **for** $j \in$ level **do**
9: **if** $u(p(j)) + u(j) < u(p(j) \cup j)$ **then**
10: $r_{p(j)} \leftarrow r_{p(j)} + r_j$
11: $l_{p(j)} \leftarrow \frac{r_{p(j)} \cdot l_{p(j)} + r_j \cdot l_j}{l_{p(j)} + l_j}$
12: $J_i \leftarrow J_i \setminus \{j\}$
13: **break for**
14: **until** $|J_i|$ decreases
▷ Deploy children ICONs of i to those $j \in J_i$ that have assigned requests
15: $D_i \leftarrow \{deploy\_to(j) : j \in J_i \setminus \{\Omega\}\}$
▷ Decide whether to terminate ICON i and handle i's children to the parent of i
16: **if** $u(p(i)) + u(\Omega) < u(p(i) \cup \Omega)$ **then**
17: $r_{p(i)} \leftarrow r_{p(i)} + r_\Omega$
18: $p(j) \leftarrow p(i), \forall j \in J_i \setminus \{\Omega\}$
19: $I \leftarrow I \setminus \{i\}$

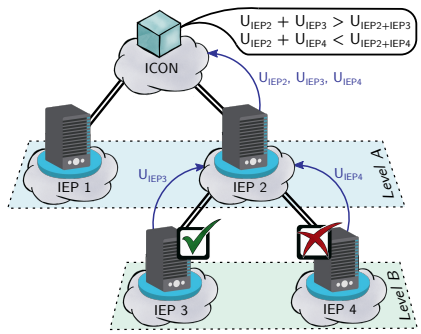

Figure 3.7: Pruning the candidate IEP tree.

Each ICON executes Algorithm 1 to choose its actions for the next time slot. First, in line 4, ICON calculates the utility of each IEP j in the discovered set of IEPs J_i as defined in Equation 3.1. Set J_i is a tree with the root IEP Ω where ICON i is currently deployed. Next, the algorithm explores the tree bottom-up pruning low utility nodes (lines 5-15). For each node, the aggregated utility of parent and child denoted as $u(p(j) \cup j)$ is calculated. In the computation of $u(p(j) \cup j)$ the requests of child j are assigned to the parent of j and served at the expenses of the parent's cost. If aggregated utility is greater than the sum of parent and child utilities, the child IEP is pruned (refer to Figure 3.7). The process iterates at the current level of the tree until no child is pruned, then the same procedure is repeated for the next level, and so on. At the end of the process, ICON deploys replicas to those IEPs that are left. Before finishing (lines 16-19), ICON performs the same utility check on itself and its parent. In the case that the aggregated utility is larger, ICON reassigns all its children, including new replicas, to its parent and terminates. Pruning the children saves the costs of operation since parents are more likely to be at aggregation points of the network topology, therefore capable of serving more requests.

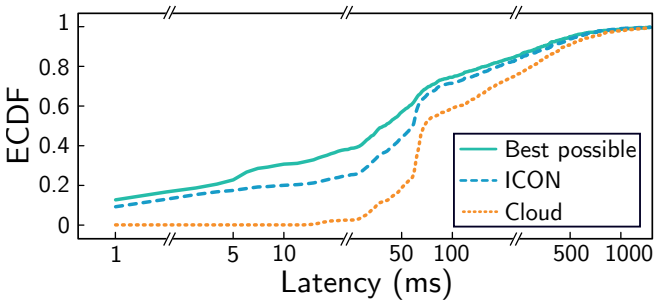

Figure 3.8: Latency ECDF of ICON overlay.

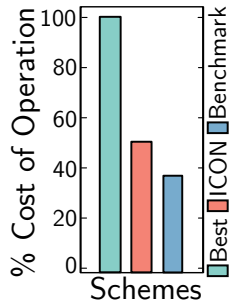

Figure 3.9: Cost of operation.

The container yard application (see Section 3.2.1) performs synchronization functions that are required for Algorithm 1 to behave correctly. For instance, when ICON decides to leave IEP, the container yard preserves the reference to its parent (and its new location, if available), so that no children will remain orphans even in the case of communication disruption.[9] It is possible that multiple ICONs discover the same IEP. In such a case, the first ICON contacting the IEP has the priority, and IEP will notify about it other ICONs attempting to make contact.

3.3.2 Evaluation

For the performance evaluation, we used the geo-distributed web trace of the World Cup 1998 event containing 1.35 billion requests [7], making it one of the largest real application traces available today. The public router dataset of CAIDA [95], limited to 2211 routers on the west coast of the US, was used to build the network topology. We assumed 20% of nodes could act as IEPs, and for economic modeling, we used AWS spot pricing distribution [213]. For the results presented below, we used equal weights for latency and cost ($w = 0.5$). Figure 3.8 displays latency ECDF for 2100 hours of the overlay's operation (the entire lifespan of the simulation). Best possible in Figure 3.8 is latency achieved with static deployment, i.e. when every IEP has an ICON deployed. The overlay's latency is close to the best possible as it manages to serve over 40% of requests below 50 ms. Figure 3.9 shows the relative costs of operation. Here *best* is again static deployment with every IEP occupied, whereas benchmark is a clairvoyant system that has complete information about the future. The overlay halves the costs of

[9] Additionally, all ICONs have reference to the origin (root of the overlay tree), so in the case of complete IEPs failure the structure of the overlay can be restored.

static deployment while loosing to benchmark around 20%. More details and results are available in Publication II and Manuscript I.

3.4 Edge Placement Strategies in the Abundance of Crowdsourced Edge Resources

Considering a futuristic outlook where crowdsourced edge resources become abundant, the question arises: what are the best locations for the placement of edge servers by CSPs and MNOs? Our attempt to address this question resulted in the Anveshak framework [139], which is included in this thesis as Publication IV.

Common edge placement approaches mainly consider the population density of potential users as a major decision criterion. Anveshak augments this model by taking into account edge resources that are likely to be available in the region. Although in the original Anveshak setting there is an alliance of CSPs and MNOs choosing cellular base stations for the edge[10] server placement, this assumption can be relaxed in the future for the more general IEP placement problem.

Figure 3.10 presents the workflow of Anveshak. During Phase I, the most active communication zones of the region are determined since these are the primary candidates for the high utilization of edge servers. To accomplish that task, Anveshak splits the region into an equally spaced grid, then utilizes cellular usage statistics to transform the grid into a heatmap of user activities. In Phase II, Anveshak incorporates potentially available edge servers (e.g., crowdsourced) into the grid data. This is accomplished by examining the locations of WiFi Access Points (APs) on the map. Anveshak associates a high density of WiFi APs with the presence of edge providers. Having the information for each grid cell concerning the user activities and potential edge presence, Anveshak proceeds to phase III, where the problem of edge server placement is reduced to Facility Location Problem [33]. The latter is known to be NP-hard, and to solve it, Anveshak restricts the size of the grid produced during Phase I, obtaining an approximate solution.

We evaluated Anveshak performance using the Telecom Italia dataset[11] and WiGLE[12] WiFi AP data. Compared to the greedy placement strategy, Anveshak achieved around 20% better edge server utilization. Publication IV encompasses more theoretical details and results of the solution.

[10] While Anveshak uses *fog* term, we unify the terminology and use *edge* instead.
[11] https://dandelion.eu/datamine/open-big-data/
[12] https://wigle.net/

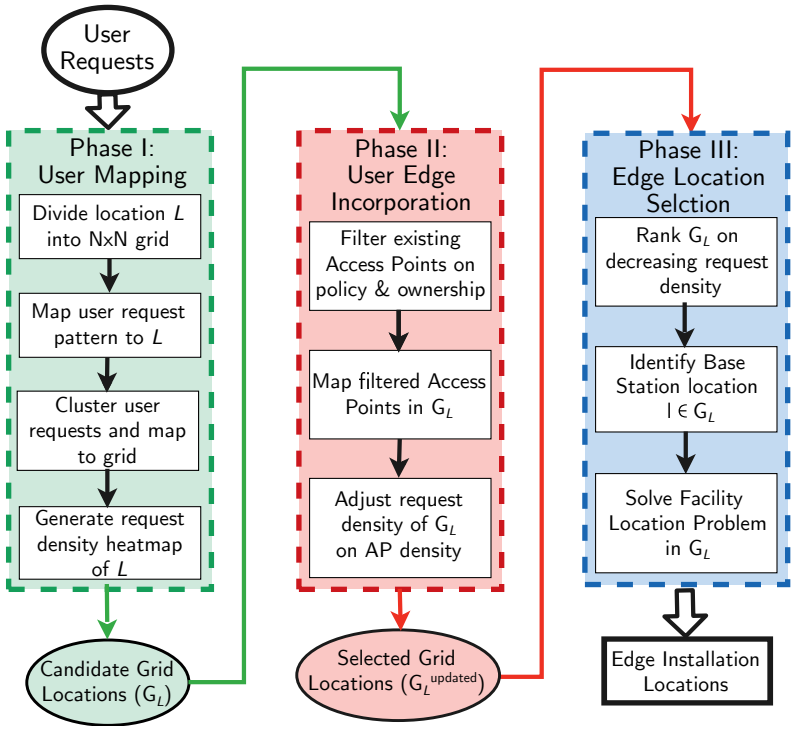

Figure 3.10: Anveshak workflow.

3.5 Designing a Market for OpenIE

As we already mentioned, OpenIE needs a market solution where participants, i.e., IEPs and their clients deploying applications, can engage each other. Such a market should be designed to spare the participants from the need for complex strategic behavior, thus improving agility. One such economic model is the Dominant Strategy Incentive Compatible (DSIC) auction, in which the best strategy for all participants is to expose their privately known valuations to the auctioneer. Such auctions are sometimes referred to as truthful, with the best-known example probably being the second-price Vickrey auction [189], in which the winner pays the second-highest bid.

Figure 3.11 presents an overview of DeCloud, the system featuring the DSIC double auction adapted specifically for DLT. DeCloud [207] is included in this thesis as Publication III. In a double auction, there are clients buying computing capacity and providers who are selling it. Since the economic mechanism is DSIC, the best strategy for both parties is to report their privately known valuations and costs.

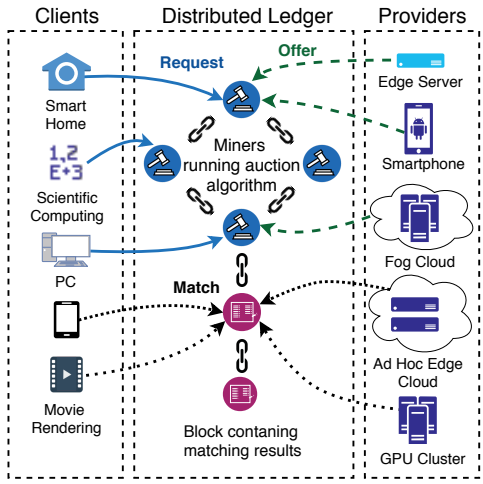

Figure 3.11: Overview of DeCloud.

As Figure 3.11 suggests, on the demand side, there are various clients in need of computing capacity, e.g., smart homes and mobile phones, the supply side is populated with providers offering resources, e.g., edge servers, crowdsourced devices, computing clusters. DeCloud is targeted for edge computing, but not limited to it, and parties interested in visual rendering or scientific computations can also use the system. The requests and offers are submitted to a P2P network of miners, which are responsible for computing the allocation, i.e., matching requests with the offers and defining the price.

Realization of DeCloud poses the following issues to solve: i) truthfulness of the auction requires that the content of bids is not exposed to the P2P network until the allocation is computed, i.e., bids must be sealed ii) matching highly heterogeneous requests and offers iii) designing a double auction mechanism satisfying DSIC property. Next, we describe how we solve these issues.

3.5.1 Two-Phase Bid Exposure Protocol

Distributed ledgers are essentially transparent[13], which in our case poses additional problems since unencrypted bids posted to an open P2P network would make the behavior of participants competitive instead of truthful. On the other hand, since the auction is decentralized, the allocation must be

[13]There are some notable exceptions, such as CryptoNote protocol or ZCash, but they are not well suited for our purpose.

publicly verifiable. Thus, bids must become open at some point. To handle the challenge, we devise a two-phase bid exposure protocol and present its operation in Figure 3.12.

During the first phase, participants submit their bids encrypted with temporary keys ensuring nondisclosure. As bids arrive, miners aggregate these encrypted bids into a new block, and, eventually, a lucky miner (miner A in Figure 3.12) will manage to obtain a PoW[14] solution for the block, securing its content cryptographically. The new block is then propagated to the rest of the miners, and the correctness of the PoW is collectively verified. Participants who identify their own bid in the block release their temporary key and the protocol enters its second phase – allocation. The content of the block can now be decrypted, and the miner computes the allocation of the auction, matching clients with providers. The results are propagated to the rest of the network, and allocated participants can confirm their commitment to the allocation. Clients can refuse since the provider they are allocated to is just the best possible match, not exactly what they have requested. At this stage, the full block is ready and sent to the rest of the network. Mining a valid block entitles the miner to the reward in digital asset-bearers, so participants are served for gratis. After the provider serves the client, counterparties can optionally update each other's reputation using some third party system[15] [90]. Additionally to the block of DeCloud, the parties might want to finalize the final agreement and payment in some third-party smart contract.

3.5.2 Matching Highly Heterogeneous Resources

DeCloud needs to handle highly heterogeneous supply and demand. The bidding language of DeCloud is capable of characterizing drastically varying needs of different clients. We achieve this by allowing clients to specify weights for the required resources to indicate their importance. We denote the significance (weight) of a type k resource as $\sigma_{(r,k)}$. Setting $\sigma_{(r,k)} = 1$ in a request signifies that the client strictly requires the specified amount of resource k. Otherwise, if $0 < \sigma_{(r,k)} < 1$, the presence of the resource is desirable, but the client can be flexible about it. In DeCloud, a resource is an inclusive concept, e.g., network location affecting latency and the reputation of a provider are also treated as resources.

In previous work, concentrating on cloud computing, which generally has a coherent resource base, similarity measures like the dot product [154]

[14] This does not need to be energy wasteful PoW but can also be PoS or another protocol.

[15] DeCloud and its protocol can also be extended to handle reputation updates.

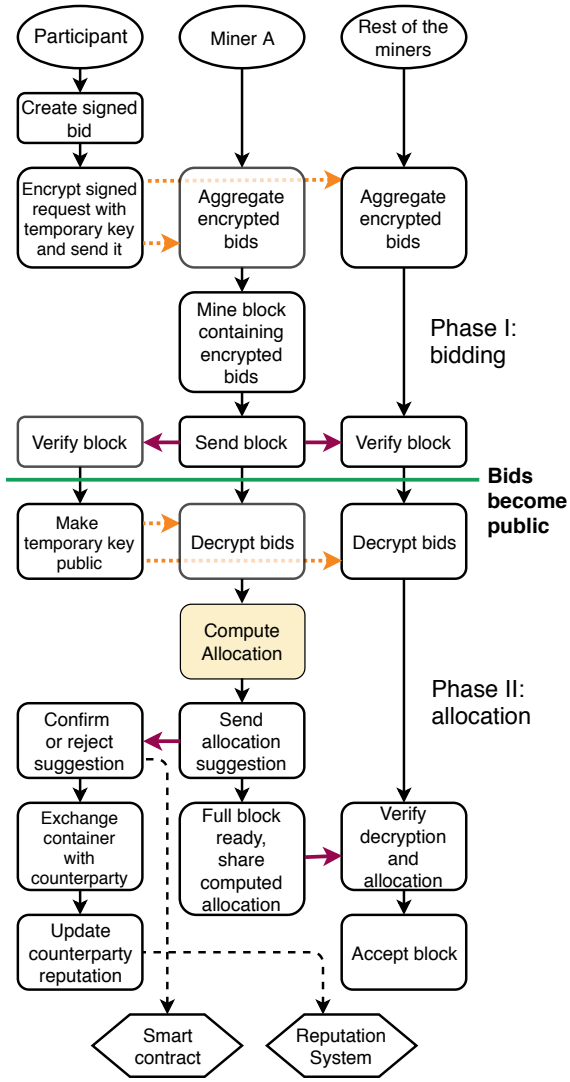

Figure 3.12: Sealed bids workflow.

is typically used. However, for flexible preferences, such an approach is challenging to adapt. For instance, if some client wants four CPU cores and one provider (A) is offering three cores and another (B) eight, then using dot product will result in choosing B, while A is a better match for the client with $\sigma_{(r,k)} < 1$. Geometric distance would work in the previous case, but if offers would be one and eight cores, then the client wanting four would be matched to one since it is the closest offer, clearly producing

the wrong output. We address the issue by attributing providers with a gravity-like force which they exert on clients, and define the *quality of match* between r and o as follows:

$$q_{(r,o)} = \sum_{k \in (K_r \cap K_o)} \sigma_{(r,k)} \frac{\rho'_{(o,k)}}{|\rho'_{(o,k)} - \rho'_{(r,k)}|^2 + 1} \tag{3.3}$$

where $\rho'_{(o,k)}$ and $\rho'_{(r,k)}$ are normalized quantities of type k resources in offer o and request r. The maximum of the scale for normalization is computed by taking the maximum value of the resource either form offers or requests included in the current block. The minimum is naturally zero. K_o and K_r are sets of all resource types occurring in offer o and request r, respectively. Equation 3.3 makes it possible to rank any offer with respect to a particular request if they have at least one common resource, i.e., $|K_r \cap K_o| > 0$.

3.5.3 Double Auction

A DeCloud auction is a continuation of the theoretical work by McAfee, who presented DSIC double auction for a single type of goods in [132]. Thus, it is convenient to start by examining McAfee's mechanism first. As McAfee's auction considers single units of identical goods, we denote a unit valuation and cost as v and c, respectively. The buyers are sorted by their valuation in descending order and sellers in ascending order by cost and paired together, as shown in Figure 3.13. We mark with index z the last pair for which condition $v_z \geq c_z$ holds. Then, either of the two is possible.

1. There is a $z+1$ pair and $p = (v_{z+1} + c_{z+1})/2 \in [c_z, v_z]$, then every participant trades at price p, see Figure 3.13.

2. Buyers pay v_z, and sellers receive payment of c_z, as shown in Figure 3.14. The pair z has to be excluded from the trade to preserve DSIC property; hence the mechanism is often called *trade reduction*.

Indeed, it is intuitively clear that if some of the participants can affect the trading price by bidding untruthfully and subsequently benefit from it, the auction is not incentive compatible (IC). Truthful bidding becomes a dominant strategy (DS) when utility of the participant can only decrease in case the bid submitted to the auctioneer is not equal to the privately known valuation, i.e., truthful bidding dominates underbidding and overbidding. We give the formal definition of DSIC property as applied for DeCloud in Publication III.

The setting of DeCloud is much more complicated than of McAfee's mechanism: goods are heterogeneous, there is no one-to-one match since

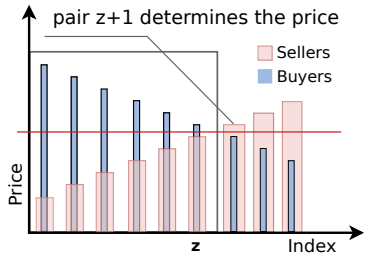

Figure 3.13: McAfee's auction, pair $z+1$ sets the trading price.

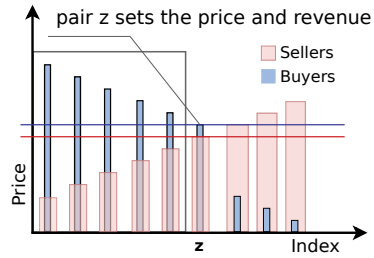

Figure 3.14: McAfee's auction, pair z sets the trading price.

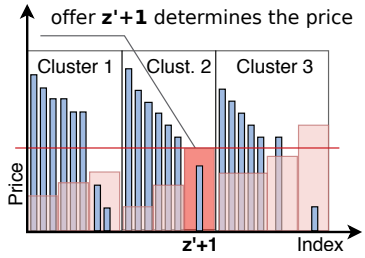

Figure 3.15: DeCloud mini-auction, pair $z+1$ sets the trading price.

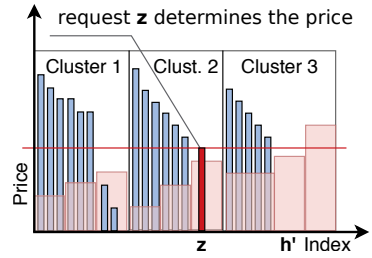

Figure 3.16: DeCloud mini-auction, pair z sets the trading price.

one offer can serve many requests, and so on. For the remedy, we use our quality of match heuristics (Equation 3.3). Applying it, we group best-matching requests and offers together in clusters. Once grouped, we can align offers and requests similarly to McAfee's mechanism and determine the trading price for each cluster. Since DeCloud is also a trade reduction mechanism, meaning that DSIC property comes at the price of excluding valid trades, we introduce the concept of *mini-auction* to alleviate the negative impact of this practice. All clusters compatible by price are grouped into mini-auctions and trade at the same price. This practice allows us to minimize the loss of valid trades to at most one per mini-auction. By price compatibility, we mean that all valid offers and requests contained in one cluster have lower costs and higher valuations than the trading price of another cluster, i.e., that price may be used for both clusters without any negative consequences. Figures 3.15 and 3.16 provide an intuitive insight for mini-auctions and determination of trading price. Above we gave a limited overview of a DeCloud auction, leaving rigorous details out of the scope of this thesis. In Publication III, we provide the formal proof that an auction is DSIC along with algorithms, definitions, and other theoretical insights.

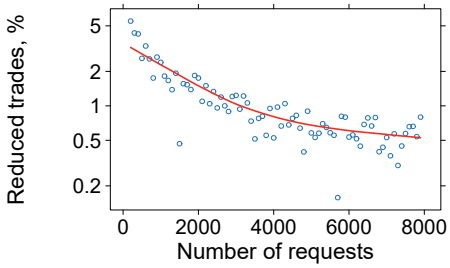

Figure 3.17: Trade reduction.

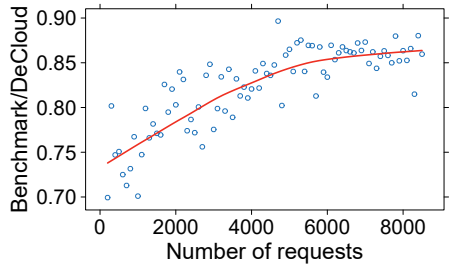

Figure 3.18: Welfare ratio.

3.5.4 Evaluation

We evaluated DeCloud performance on real-world data and used Google Cluster Data [80] for request modeling, whereas for pricing – Amazon EC2 M5 instance types [3]. Logically, since truthfulness comes at the price of excluding valid trades, our main purpose was to investigate the adverse influence of DSIC implementation. Our benchmark was a double auction with exactly the same algorithm, however, without DSIC, thus capable to achieve the best welfare[16], assuming that participants continue to bid truthfully. As Figure 3.17 suggests, the percentage of reduced trades is up to 5% in small markets, falling below 1% when the number of requests exceeds 5000. Figure 3.18 displays the ratio between the welfare of DeCloud and the benchmark. DeCloud achieves 75% of the best possible welfare despite the minimal size of the market. In more favorable market conditions, the welfare ratio rises to 85%. In Figures 3.17 and 3.18 we draw the Loess curves to highlight the trends. Given the benefits that the DSIC auction brings, e.g., no need for complex market strategy, reduced risk of market manipulation, the marginal decrease in welfare along with below 5% of reduced trades is likely to be an acceptable tradeoff. Publication III contains more details and results.

3.6 Distributed Ledger Technologies in OpenIE

In OpenIE, we often considered DLT as a way to handle agreements and transactions in a distributed fashion (Publication I and Publication II), improving overall agility. In Publication III, we have shown that it is possible and convenient to build a distributed auction on top of DLT. Our

[16]For double auction, welfare is defined as a difference between valuations and costs of allocated buyers and sellers, respectively [115].

work [210], which is included in this thesis as Publication V, examines the state-of-the-art of distributed ledgers and their applicability for the needs of OpenIE.

3.6.1 The Ricardian Triple

We have already introduced smarts contracts and examined their applications for edge and cloud computing in Section 2.2. However, there is another important technique to handle the digital agreement, namely, the Ricardian Contract [31, 84]. The purpose of the Ricardian Contract is to put the text of a legal agreement in tamper-proof digital form, make it machine-readable, and provide a convenient way to reference it in digital transactions. Contrary to smart contracts, the role of which is execution and enforcement of some implicit commitment between participants, the role of the Ricardian Contract is to represent and protect the legal agreement itself. Thus, the Ricardian and smart contract form a kind of a symbiotic relation, resulting in the Ricardian triple [39, 85]: <prose, parameters, code>, where the prose is the Ricardian Contract, i.e., digitalized legal agreement, the code is smart contract enforcing of the prose, and introduction of parameters provides the flexibility required in the binding of formulation and enforcement together. The triple acts like a template where parameters actualize the specific details of a deal. We consider the Ricardian triple as a solid foundation for building the contractual framework of OpenIE. For instance, the Ricardian Contract can be used for SLA representation, while a smart contract for the practical engagement of counterparties. The details of the agreement, such as latency, price, etc. can be customized with parametrization.

3.6.2 The Requirements of OpenIE

The desiderata of OpenIE for the DLT platforms encompasses at least the following:

1. Environmentally sustainable operation. As Figure 3.19 suggests for the year 2018, Bitcoin consumes as much energy as Portugal, while Ethereum nears Costa Rica.

2. Speed of transaction confirmation. The dynamicity of edge computing environments can vary, e.g., for ICON, the plausible speed can be on the order of magnitude of seconds.

3. Cost of transaction. The cost needs to be significantly lower than the deployment cost itself, e.g., AWS VM hourly prices [4] may start as low as $0.011.

4. Protection against financial volatility. The volatility (see Figure 3.20) of digital currencies used by DLT poses significant risk and can be a threat to DLT adaption.

5. Privacy of transactions. The transparency of ledgers such as Ethereum [196] can have adverse effects on business.

6. Accountability of participants. Public DLT platforms, although not designed for privacy, preserve a good level of participants' anonymity. However, OpenIE would also require accountability from the participants, facing anonymity vs. accountability dilemma [123], which is typical for open systems.

7. Escrow service and conflict resolution. The payments should be transferred only in the case of appropriate service delivery by IEP, therefore escrow service is needed. Additionally, there might be disagreement, e.g., on whether the SLA was fulfilled or not. The resolution of such conflicts might employ the usage of a reputation system, and DLT provide good opportunities to store reputation data securely.

8. Legislation for digital agreement. The digital agreement must have a legislative status to be effective, and there are initiatives to grant such status to DLT [22]. However, more needs to be done.

9. Truly distributed operation without a central authority. Popular platforms are getting more and more centralized [126], dismantling the core idea of central authority absence.

Next, we discuss which of the above upcoming DLT are already being addressed, and what remains a future challenge.

3.6.3 The Promises and Challenges of DLT

The paramount issue of DLT has been sustainability and scalability. Fortunately, there is a new generation of systems abandoning the usage of PoW, which has been the performance bottleneck and the reason for high energy consumption. Many new systems utilize variants of the Proof-of-Stake (PoS) protocol, and also Ethereum is planning the migration to PoS [56]. Overly simplified, in PoS, peers who hold more assets of the system have higher chances to generate a new block and gain the reward. The more

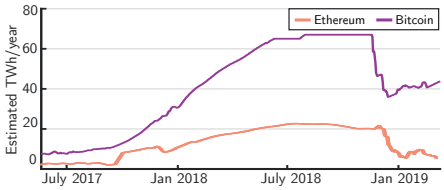

Figure 3.19: Bitcoin and Ethereum energy consumption [46].

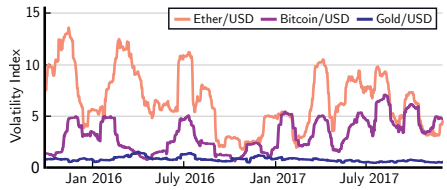

Figure 3.20: Volatility of Ether, Bitcoin, and gold vs. US dollar [23].

some entity has at stake, the more trusted it is. Logically, PoS employs advanced techniques in an attempt to avoid the rich get richer loop. PoS consumes a minimal amount of energy, while radically improving performance, e.g., transaction throughput. Most recent systems use Delegated Proof-of-Stake (DPoS) [118], which uses PoS to elect block producers for the next round.

The characteristics of some popular systems with smart contract support is summarized in Table 3.1. Compared to other systems, Ethereum has high transaction costs, while EOS has no fees at all. However, Ethereum has a large support base among miners, while [200] presents the opinion that EOS is actually a centralized system. Thus, requirements 1, 2, and 3 can be satisfied by present systems, allegedly, with a risk of increased centralization.

To address the volatility of cryptocurrencies, there are stablecoins, e.g., Tether [186], the exchange rate of which is fixed to US dollars. However, buying US dollars to guarantee stability of the exchange rate is handled by Tether Ltd. private company. Thus, Tether contradicts the decentralization idea of Bitcoin-like cryptocurrencies, and one owning Tethers needs to trust Tether Ltd explicitly. There is a controversy over how accurate Tether Ltd. is in handling its duties [190]. To conclude, our requirement 4 implies trust in private companies or other organizations.

For competitive reasons, IEPs or other business entities involved in OpenIE might want to hide their pricing or other sensitive information. In Ethereum or EOS, it is difficult to accomplish since all data is transparent by default. There are platforms attempting to address the issue, e.g., DERO Project [160] is developing secure private smart contracts. However, the support of the community remains quite limited as of now. So, given that there will be reliable platforms like DERO with broad miner support, requirement 5 can be addressed in the future.

The problem of accountability (requirement 6) brings up centralization and privacy concerns. Technically, the problem is solved for permissioned

distributed ledgers, where only authorized participants can join. For open blockchain systems, this might involve some compromises, e.g., usage of third-party authentication and identity verification services.

Technical aspects of requirement 7 are well handled by smart contracts [198], whereas conflict resolution procedures are related to the legislation of digital agreement (requirement 8), where much of progress is expected to happen in the future.

To summarize, platforms like EOS could handle many of the tasks that OpenIE would impose even today. However, there is no distributed ledger that would satisfy all of our requirements. Nevertheless, the flamboyant development of the field and extensive community support for Bitcoin and Ethereum make a positive outlook for the future. Interledgers appear especially promising, e.g., Cosmos[17] or Polkadot[18], since they enable seamless interaction between various distributed ledger platforms so that entities favoring different systems can interact without changes in their preferences. Many of the requirements we presented involve the centralization vs. decentralization dilemma and addressing some of them might involve a compromise solution. At present, Ethereum is the largest public open distributed ledger supporting smart contacts, and the future of DLT is likely to be influenced significantly by how successful the transition to PoS-based Ethereum 2.0 will eventually be.

[17] https://cosmos.network/
[18] https://polkadot.network/

Platform \ Metric	Transactions per second (TPS)	Transaction confirmation time	Transaction fee	Number of nodes	Consensus algorithm
Ethereum	10-30	6 min	$0.35	8717	PoW, moving to PoS
EOS	3000	1.5 sec	$0	21 (528)	DPoS
Stellar	1000	4-5 sec	0.00001 XLM†	434	SCP
TRON	748	5 min	$0	1058	DPoS
Cardano	50-250	5 min	0.155381 ADA‡	NA	PoS (Ouroboros)

†XLM is Stellar Lumen token, ‡ADA is Cardano token.

Table 3.1: Performance of platforms supporting smart contracts. Numerical data as of March 2019.

Chapter 4

Conclusion

This thesis introduced Open Infrastructure for Edge (OpenIE) and devised solutions for its major components. The motivation of OpenIE is to make edge computing a pervasive paradigm on a par with cloud computing. In this chapter, we look again at the research questions formulated in Section 1.2, contemplating the deficiencies of the devised solutions and set the directions for future work.

4.1 Research Questions Revisited

RQ1. *Is a global-scale discovery mechanism for edge resources feasible, and how should it operate?*

We found that existing DNS provides excellent opportunities for the discovery of edge resources, and in Section 3.2, we presented ExEC – a solution enabling the discovery of IEPs on a global scale. For its operation, ExEC requires no changes to the existing infrastructure or networking protocols. The only requirement for edge providers is to register themselves in the DNS, and thereafter they can be located on the fly by anyone interested in deploying services at the edge of the network.

RQ2. *How can autonomous decision making tame the dynamicity required of edge computing, and how can it be incorporated into practical systems?*

As edge computing supposes highly dynamic environments, e.g., where users migrate or flash crowds discovering new applications emerge, it would be convenient if the underlying infrastructure would be capable of handling as spatiotemporal shifts in users' behavior, as well

as the automated deployment of applications. We proposed an overlay of Intelligent Containers (ICONs), which are entities capable of autonomous decision making based on the observations of their environment. Capable of discovering edge providers by using the ExEC discovery mechanism, ICON can bring the edge service it encapsulates to the location where it is needed the most by its users.

RQ3. *How will the presence of IEPs affect the placement of edge servers?*

We designed the Anveshak platform, which placement strategy takes into account not only the density of potential users but also existing deployments of crowdsourced and other edge computing resources.

RQ4. *What market model would be beneficial for edge providers and application owners to enter into the trade, and how can such a market be technically implemented?*

As such a market must be agile and spare the participants from the need for complex strategy, we developed DeCloud – Dominant Strategy Incentive Compatible (DSIC) double auction mechanism adapted for decentralized execution on a distributed ledger. In a DSIC double auction, the dominant strategy for buyers and sellers is to submit their privately known valuations and costs to the auctioneer, which then matches them together and determines trading prices. Such a design of the trading mechanism reduces the chances of market manipulation attempts and simplifies the behavior of counterparties.

RQ5. *What potential existing DLT platforms have to provide agreement and transaction handling services for OpenIE?*

Since it is convenient if integral parts of OpenIE would utilize distributed agreement and transaction handling, we examine the maturity of existing DLT solutions to take up such a role. We find that the upcoming generation of platforms, e.g., utilizing PoS consensus algorithms, are environmentally friendly and performant, as opposed to PoW-based pioneers of blockchain. There are still problems related to the high volatility of cryptocurrencies, accountability, handling sensitive data in smart contracts, etc. A solution to some of the existing issues would require a tradeoff in which the decentralization offered by DLT would suffer to some extent. Nevertheless, the broad support of DLT by the community and the fast development of technology gives a reason for optimism. Ethereum, the largest platform supporting smart contracts, is planning the migration from PoW to PoS-based protocol. In the case of success, DLT are likely

to get a further stimulus for technological evolution and adaption in the economy, including edge computing.

4.2 Future Work

This thesis has proposed the concept of OpenIE and offered technical solutions for its major features. However, given the scale of the undertaking, we presented just a scratch on a surface. To become a reality, OpenIE needs a lot of technical development along with accompanying theoretical work. We continue the development of the ICON overlay, aiming to make it an operational system that edge providers can utilize. Also, container yard software and related standardization require additional development efforts. We look for the improvements of the algorithmic solution, although the current approach achieves good results. DeCloud has been majorly a theoretical effort, and we are planning to implement it as a real blockchain system. The tendermint protocol[1] supporting inter-blockchain communication is particularly attractive since it allows clients to use the DLT platforms of their choice. Then, it would be compelling to evaluate how ICONs would perform using DeCloud as a market place. Our discovery solution, ExEC, can be improved in several ways. Particularly, limitations of on-path discovery can be addressed by extending observations to nearby domains of clients, whereas for the alleviation of traceroute's performance overhead networking topology maps can be used. Even in the case the idea of DNS registration will not get common among edge providers, it will not dismantle OpenIE since DeCloud allows to specify a location in its offers and requests, thus providing an alternative for ExEC discovery. Many security issues require additional investigation before the global-scale environment where self-optimizing services that migrate freely can become a reality.

[1] https://tendermint.com/

References

[1] Nasir Abbas, Yan Zhang, Amir Taherkordi, and Tor Skeie. Mobile edge computing: A survey. *IEEE Internet of Things Journal*, 5(1):450–465, 2017.

[2] Mohammed A AlZain, Eric Pardede, Ben Soh, and James A Thom. Cloud computing security: from single to multi-clouds. In *2012 45th Hawaii International Conference on System Sciences*, pages 5490–5499. IEEE, 2012.

[3] Amazon. Amazon EC2 prices. "`https://aws.amazon.com/pricing`", 2018.

[4] Amazon. Amazon EMR Pricing. "`https://aws.amazon.com/emr/pricing/`", 2018.

[5] Ganesh Ananthanarayanan, Paramvir Bahl, Peter Bodík, Krishna Chintalapudi, Matthai Philipose, Lenin Ravindranath, and Sudipta Sinha. Real-time video analytics: The killer app for edge computing. *computer*, 50(10):58–67, 2017.

[6] D. P. Anderson. Boinc: a system for public-resource computing and storage. In *Fifth IEEE/ACM International Workshop on Grid Computing*, pages 4–10, Nov 2004.

[7] Martin Arlitt and Tai Jin. A workload characterization study of the 1998 world cup web site. *IEEE network*, 14(3):30–37, 2000.

[8] ARM. ARM TrustZone. "`https://www.arm.com/products/security-on-arm/trustzone`", 2020.

[9] Austin Aske and Xinghui Zhao. Supporting multi-provider serverless computing on the edge. In *Proceedings of the 47th International Conference on Parallel Processing Companion*, pages 1–6, 2018.

[10] AWS. Aws iot for the edge. "https://aws.amazon.com/iot/solutions/iot-edge/", 2020.

[11] AWS. Aws outposts. "https://aws.amazon.com/outposts/", 2020.

[12] AWS. Lambda. "https://aws.amazon.com/lambda/", 2020.

[13] Randall E Bailey, Jarvis James Arthur III, and Steven P Williams. Latency requirements for head-worn display s/evs applications. In *Enhanced and Synthetic Vision 2004*, volume 5424, pages 98–109. International Society for Optics and Photonics, 2004.

[14] Ioana Baldini, Paul Castro, Kerry Chang, Perry Cheng, Stephen Fink, Vatche Ishakian, Nick Mitchell, Vinod Muthusamy, Rodric Rabbah, Aleksander Slominski, et al. Serverless computing: Current trends and open problems. In *Research Advances in Cloud Computing*, pages 1–20. Springer, 2017.

[15] Luciano Baresi and Danilo Filgueira Mendonça. Towards a serverless platform for edge computing. In *2019 IEEE International Conference on Fog Computing (ICFC)*, pages 1–10. IEEE, 2019.

[16] Luciano Baresi, Danilo Filgueira Mendonça, and Martin Garriga. Empowering low-latency applications through a serverless edge computing architecture. In *European Conference on Service-Oriented and Cloud Computing*, pages 196–210. Springer, 2017.

[17] Jeff Barr. Firecracker lightweight virtualization for serverless computing. "https://www.zdnet.com/article/10-edge-computing-vendors-to-watch/", 2018.

[18] Suzan Bayhan, Anatolij Zubow, and Adam Wolisz. Spass: Spectrum sensing as a service via smart contracts. In *2018 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pages 1–10. IEEE, 2018.

[19] Pronaya Bhattacharya, Sudeep Tanwar, Rushabh Shah, and Akhilesh Ladha. Mobile edge computing-enabled blockchain framework—a survey. In *Proceedings of ICRIC 2019*, pages 797–809. Springer, 2020.

[20] BOINC. Compute for Science. "https://boinc.berkeley.edu/", 2020.

[21] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. Fog computing and its role in the internet of things. In *Proceedings*

of the first edition of the MCC workshop on Mobile cloud computing, pages 13–16. ACM, 2012.

[22] Jordan Buie. Lawmaker's bill would accept technology behind Bitcoin for contracts in Tennessee. "`https://eu.tennessean.com/story/news/2018/02/06/bitcoin-law-tennessee-cryptocurrency-bill/311641002/`", 2018.

[23] Buy Bitcoin Worldwide. The Bitcoin Volatility Index. "`https://www.buybitcoinworldwide.com/volatility-index/`", 2019.

[24] X. Cao, J. Xu, and R. Zhang. Mobile edge computing for cellular-connected uav: Computation offloading and trajectory optimization. In *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 1–5, June 2018.

[25] Alejandro Cartas, Martin Kocour, Aravindh Raman, Ilias Leontiadis, Jordi Luque, Nishanth Sastry, Jose Nuñez-Martinez, Diego Perino, and Carlos Segura. A reality check on inference at mobile networks edge. In *Proceedings of the 2nd International Workshop on Edge Systems, Analytics and Networking*, pages 54–59, 2019.

[26] Abhishek Chandra, Jon Weissman, and Benjamin Heintz. Decentralized edge clouds. *IEEE Internet Computing*, 17(5):70–73, 2013.

[27] Lucas Chaufournier, Prateek Sharma, Franck Le, Erich Nahum, Prashant Shenoy, and Don Towsley. Fast transparent virtual machine migration in distributed edge clouds. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, page 10. ACM, 2017.

[28] Hosting Checker. Reverse ip lookup. "`https://hostingchecker.com/tools/reverse-ip-lookup`", 2020.

[29] Youdong Chen, Qiangguo Feng, and Weisong Shi. An industrial robot system based on edge computing: An early experience. In {*USENIX*} *Workshop on Hot Topics in Edge Computing (HotEdge 18)*, 2018.

[30] Bin Cheng, Apostolos Papageorgiou, and Martin Bauer. Geelytics: Enabling on-demand edge analytics over scoped data sources. In *2016 IEEE International Congress on Big Data (BigData Congress)*, pages 101–108. IEEE, 2016.

[31] Usman W. Chohan. What Is a Ricardian Contract? *SSRN*, 2017.

[32] Aakanksha Chowdhery, Marco Levorato, Igor Burago, and Sabur Baidya. Urban iot edge analytics. In *Fog computing in the internet of things*, pages 101–120. Springer, 2018.

[33] Richard Church and Charles R Velle. The maximal covering location problem. *Papers in regional science*, 32(1):101–118, 1974.

[34] Franco Cicirelli, Antonio Guerrieri, Giandomenico Spezzano, and Andrea Vinci. An edge-based platform for dynamic smart city applications. *Future Generation Computer Systems*, 76:106–118, 2017.

[35] CISCO. Cisco Annual Internet Report (20182023) White Paper. "`https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html`", 2020.

[36] CISCO. Cisco IoT. "`https://www.cisco.com/c/en/us/solutions/internet-of-things/overview.html`"", 2020.

[37] CISCO. Cisco IR829 Industrial Integrated Services Routers. "`https://www.cisco.com/c/en/us/products/collateral/routers/829-industrial-router/datasheet-c78-734981.pdf`"", 2020.

[38] CISCO. Cisco Unified Computing. "`https://www.cisco.com/c/en/us/products/servers-unified-computing`"", 2020.

[39] Christopher D Clack, Vikram A Bakshi, and Lee Braine. Smart contract templates: foundations, design landscape and research directions. *arXiv preprint arXiv:1608.00771*, 2016.

[40] Vittorio Cozzolino, Aaron Yi Ding, and Jörg Ott. Fades: Fine-grained edge offloading with unikernels. In *Proceedings of the Workshop on Hot Topics in Container Networking and Networked Systems*, pages 36–41, 2017.

[41] Vittorio Cozzolino, Aaron Yi Ding, and Jörg Ott. Edge chaining framework for black ice road fingerprinting. In *Proceedings of the 2nd International Workshop on Edge Systems, Analytics and Networking*, pages 42–47, 2019.

[42] Antonio Cuomo, Giuseppe Di Modica, Salvatore Distefano, Antonio Puliafito, Massimiliano Rak, Orazio Tomarchio, Salvatore Venticinque, and Umberto Villano. An SLA-based broker for cloud infrastructures. *Journal of grid computing*, 11(1):1–25, 2013.

[43] D2iQ. Distributed cloud operating system. "https://dcos.io/", 2020.

[44] A. Davis, J. Parikh, and W. E. Weihl. Edge computing: Extending enterprise applications to the edge of the internet. In *Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers &Amp; Posters*, WWW Alt. '04, pages 180–187, New York, NY, USA, 2004. ACM.

[45] Eyal de Lara, Carolina S Gomes, Steve Langridge, S Hossein Mortazavi, and Meysam Roodi. Hierarchical serverless computing for the mobile edge. In *2016 IEEE/ACM Symposium on Edge Computing (SEC)*, pages 109–110. IEEE, 2016.

[46] Digiconomist. Ethereum Energy Consumption Index. "https://digiconomist.net/ethereum-energy-consumption", 2019.

[47] Hoang T Dinh, Chonho Lee, Dusit Niyato, and Ping Wang. A survey of mobile cloud computing: architecture, applications, and approaches. *Wireless communications and mobile computing*, 13(18):1587–1611, 2013.

[48] Docker. Docker containers. "https://www.docker.com/", 2019.

[49] Caroline Donnelly. Aws fleshes out edge computing strategy with hyper-local datacentre hubs planned in major cities. "https://www.computerweekly.com/news/252474948/AWS-fleshes-out-edge-computing-strategy-with-hyper-local-datacentre-hubs-planned-in-major-cities", 2020.

[50] Utsav Drolia, Rolando Martins, Jiaqi Tan, Ankit Chheda, Monil Sanghavi, Rajeev Gandhi, and Priya Narasimhan. The case for mobile edge-clouds. In *2013 IEEE 10th International Conference on Ubiquitous Intelligence and Computing and 2013 IEEE 10th International Conference on Autonomic and Trusted Computing*, pages 209–215. IEEE, 2013.

[51] Edge Network. Edge. "https://edge.network", 2020.

[52] Edge Network. Explorer. "https://explorer.edge.network/", 2020.

[53] D2iQ Edward Hsu. Announcing dc/os 1.11: Edge & multi-cloud operations now a reality. "https://d2iq.com/blog/dcos-1_11-overview/", 2018.

[54] Elijah and Gabriel Open Source Software. "http://elijah.cs.cmu.edu/development.html"", 2020.

[55] Melike Erol-Kantarci and Sukhmani Sukhmani. Caching and computing at the edge for mobile augmented reality and virtual reality (ar/vr) in 5g. In *Ad Hoc Networks*, pages 169–177. Springer, 2018.

[56] Ethereum. Ethereum 2.0 (Serenity). "https://github.com/ethereum/eth2.0-specs", 2020.

[57] ETSI. Multi-access Edge Computing (MEC); Framework and Reference Architecture. GS MEC 003 V2.1.1, 2019.

[58] ETSI. Multi-access Edge Computing. "https://www.etsi.org/technologies/multi-access-edge-computing", 2020.

[59] Dave Evans. The Internet of Things: how the next evolution of the internet is changing everything (april 2011). *White Paper by Cisco Internet Business Solutions Group (IBSG)*, 2012.

[60] Qiang Fan and Nirwan Ansari. Workload allocation in hierarchical cloudlet networks. *IEEE Communications Letters*, 22(4):820–823, 2018.

[61] T. M. Fernndez-Carams and P. Fraga-Lamas. A review on the use of blockchain for the internet of things. *IEEE Access*, 6:32979–33001, 2018.

[62] Mohamed Amine Ferrag, Makhlouf Derdour, Mithun Mukherjee, Abdelouahid Derhab, Leandros Maglaras, and Helge Janicke. Blockchain technologies for the internet of things: Research issues and challenges. *IEEE Internet of Things Journal*, 6(2):2188–2204, 2018.

[63] Huber Flores, Pan Hui, Sasu Tarkoma, Yong Li, Satish Srirama, and Rajkumar Buyya. Mobile code offloading: from concept to practice and beyond. *IEEE Communications Magazine*, 53(3):80–88, 2015.

[64] Forbes. How much data do we create every day? the mind-blowing stats everyone should read. "https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/"", 2018.

[65] Conner Forrest. Ten edge computing vendors to watch. "https://aws.amazon.com/blogs/aws/firecracker-lightweight-virtualization-for-serverless-computing/", 2018.

[66] Apache Software Foundation. Mesos. "https://mesos.apache.org/", 2020.

[67] The Ethereum Foundation. Ethereum smart contract. "https://www.ethereum.org/", 2019.

[68] The Linux Foundation. Kernel virtual machine. "http://www.linux-kvm.org", 2018.

[69] The Linux Foundation. Hyperledger smart contract. "https://www.hyperledger.org/", 2019.

[70] The Linux Foundation. LXC linux containers. "http://lxc.sourceforge.net", 2019.

[71] Linton C Freeman. A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41, 1977.

[72] Weichao Gao, William G Hatcher, and Wei Yu. A survey of blockchain: techniques, applications, and challenges. In *2018 27th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–11. IEEE, 2018.

[73] Julien Gedeon, Florian Brandherm, Rolf Egert, Tim Grube, and Max Mühlhäuser. What the fog? edge computing revisited: Promises, applications and future challenges. *IEEE Access*, 7:152847–152878, 2019.

[74] Julien Gedeon, Jeff Krisztinkovics, Christian Meurisch, Michael Stein, Lin Wang, and Max Mühlhäuser. A multi-cloudlet infrastructure for future smart cities: An empirical study. In *Proceedings of the 1st International Workshop on Edge Systems, Analytics and Networking*, pages 19–24, 2018.

[75] Ali Gerrard. What is kubernetes? an introduction to the wildly popular container orchestration platform. "https://blog.newrelic.com/engineering/what-is-kubernetes/", 2019.

[76] Ahmed G Ghandour, Mohamed Elhoseny, and Aboul Ella Hassanien. Blockchains for smart cities: a survey. In *Security in Smart Cities: Models, Applications, and Challenges*, pages 193–210. Springer, 2019.

[77] Alex Glikson, Stefan Nastic, and Schahram Dustdar. Deviceless edge computing: extending serverless computing to the edge of the network. In *Proceedings of the 10th ACM International Systems and Storage Conference*, pages 1–1, 2017.

[78] Golem. Global Network Data. "https://stats.golem.network/ui/", 2020.

[79] Golem. Worldwide Supercomputer. "https://golem.network/", 2020.

[80] Google. Cluster Data. "https://github.com/google/cluster-data", 2011.

[81] Google. Google cloud iot. "https://cloud.google.com/solutions/iot", 2020.

[82] Barry Greene. 5g latency reality checks. "https://www.senki.org/5g-latency-reality-checks/", 2018.

[83] Gridcoin. "https://gridcoin.us/", 2020.

[84] I. Grigg. The ricardian contract. In *Proceedings. First IEEE International Workshop on Electronic Contracting, 2004.*, pages 25–31, July 2004.

[85] I. Grigg. The sum of all chains lets converge! "http://financialcryptography.com/mt/archives/001556.html", 2015.

[86] Joo Guerreiro, Rui Moura, and Joo Nuno Silva. Teender: Sgx enclave migration using hsms. *Computers & Security*, page 101874, 2020.

[87] Kiryong Ha and Mahadev Satyanarayanan. Openstack++ for cloudlet deployment. *School of Computer Science Carnegie Mellon University Pittsburgh*, 2015.

[88] Toshiyuki Haramaki, Akihito Yatsuda, and Hiroaki Nishino. A robot assistant in an edge-computing-based safe driving support system. In *International Conference on Network-Based Information Systems*, pages 144–155. Springer, 2018.

[89] Najmul Hassan, Kok-Lim Alvin Yau, and Celimuge Wu. Edge computing in 5g: A review. *IEEE Access*, 7:127276–127289, 2019.

[90] Ferry Hendrikx, Kris Bubendorfer, and Ryan Chard. Reputation systems: A survey and taxonomy. *Journal of Parallel and Distributed Computing*, 75:184–197, 2015.

[91] Benjamin Hindman, Andy Konwinski, Matei Zaharia, Ali Ghodsi, Anthony D Joseph, Randy H Katz, Scott Shenker, and Ion Stoica.

Mesos: A Platform for Fine-Grained Resource Sharing in the Data Center . In *NSDI*, volume 11, pages 22–22, 2011.

[92] Eric J Horvitz, Harold L Cochrane, Rene A Vega, and Angel S Calvo. Cloud computing resource broker, 2012. US Patent 8,244,559.

[93] Mahmood Hosseini, Constantinos Marios Angelopoulos, Wei Koong Chai, and Stephane Kundig. Crowdcloud: a crowdsourced system for cloud infrastructure. *Cluster Computing*, 22(2):455–470, 2019.

[94] Long Hu, Yiming Miao, Gaoxiang Wu, Mohammad Mehedi Hassan, and Iztok Humar. irobot-factory: An intelligent robot factory based on cognitive manufacturing and edge computing. *Future Generation Computer Systems*, 90:569–577, 2019.

[95] Bradley Huffaker, Marina Fomenkov, et al. Internet topology data comparison. Technical report, Cooperative Association for Internet Data Analysis (CAIDA), 2012.

[96] IEEE. 1934-2018 - IEEE Standard for Adoption of OpenFog Reference Architecture for Fog Computing. "`https://standards.ieee.org/standard/1934-2018.html`"", 2018.

[97] Internet Engineering Task Force (IETF). DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION - RFC 1035. "`https://tools.ietf.org/html/rfc1035`", 1987.

[98] Internet Engineering Task Force (IETF). Common DNS Operational and Configuration Errors - RFC 1912. "`https://tools.ietf.org/html/rfc1912`", 1996.

[99] Internet Engineering Task Force (IETF). DNS SRV Record - RFC 2782. "`https://tools.ietf.org/rfc/rfc2782.txt`", 2000.

[100] iExec. "`https://iex.ec/`", 2020.

[101] Takayuki Imada. Mirageos unikernel with network acceleration for iot cloud environments. In *Proceedings of the 2018 2nd International Conference on Cloud and Big Data Computing*, pages 1–5, 2018.

[102] Industrial Internet Consortium. OpenFog Initiative. "`https://www.iiconsortium.org/index.htm`", 2019.

[103] Intel Software Guard eXtensions. "`https://software.intel.com/en-us/sgx`", 2020.

[104] Razi Iqbal, Talal Ashraf Butt, M Omair Shafique, Manar Wasif Abu Talib, and Tariq Umer. Context-aware data-driven intelligent framework for fog infrastructures in internet of vehicles. *IEEE Access*, 6:58182–58194, 2018.

[105] Bukhary Ikhwan Ismail, Ehsan Mostajeran Goortani, Mohd Bazli Ab Karim, Wong Ming Tat, Sharipah Setapa, Jing Yuan Luke, and Ong Hong Hoe. Evaluation of docker as edge computing platform. In *2015 IEEE Conference on Open Systems (ICOS)*, pages 130–135. IEEE, 2015.

[106] Joab Jackson. Mesosphere 1.11 focuses on edge computing, multicloud and disaster recovery. "`https://thenewstack.io/mesosphere-1-11-focuses-edge-computing-disaster-recovery/`", 2018.

[107] Pooyan Jamshidi, Claus Pahl, Nabor C Mendonça, James Lewis, and Stefan Tilkov. Microservices: The journey so far and challenges ahead. *IEEE Software*, 35(3):24–35, 2018.

[108] Changkun Jiang, Lin Gao, Tong Wang, Yufei Jiang, and Jianqiang Li. Crowd-mecs: A novel crowdsourcing framework for mobile edge caching and sharing, 2020.

[109] Wolfgang John, Joacim Halén, Xuejun Cai, Chunyan Fu, Torgny Holmberg, Vladimir Katardjiev, Mina Sedaghat, Pontus Sköldström, Daniel Turull, Vinay Yadhav, et al. Making cloud easy: design considerations and first components of a distributed operating system for cloud. In *10th {USENIX} Workshop on Hot Topics in Cloud Computing (HotCloud 18)*, 2018.

[110] Eric Jonas, Johann Schleier-Smith, Vikram Sreekanti, Chia-Che Tsai, Anurag Khandelwal, Qifan Pu, Vaishaal Shankar, Joao Carreira, Karl Krauth, Neeraja Yadwadkar, et al. Cloud programming simplified: A berkeley view on serverless computing. *arXiv preprint arXiv:1902.03383*, 2019.

[111] Emiko Jozuka. How high speed communications networks are making remote surgery realistic. "`https://www.vice.com/en_us/article/ezp4qm/how-high-speed-communications-networks-are-making-remote-surgery-realistic`", 2015.

[112] James Kempf, Sambit Nayak, Remi Robert, Jim Feng, Kunal Rajan Deshmukh, Anshu Shukla, Aleksandra Obeso Duque, Nanjangud

Narendra, and Johan Sjöberg. The nubo virtual services marketplace. *arXiv preprint arXiv:1909.04934*, 2019.

[113] Wazir Zada Khan, Ejaz Ahmed, Saqib Hakak, Ibrar Yaqoob, and Arif Ahmed. Edge computing: A survey. *Future Generation Computer Systems*, 97:219 – 235, 2019.

[114] Zaheer Khan, Abdul Ghafoor Abbasi, and Zeeshan Pervez. Blockchain and edge computing–based architecture for participatory smart city applications. *Concurrency and Computation: Practice and Experience*, page e5566, 2019.

[115] Vijay Krishna. *Auction theory*. Academic press, 2009.

[116] Stéphane Jean-Jacques Kuendig, Jose Rolim, Konstantinos Marios Angelopoulos, and Mahmood Hosseini. Crowdsourced edge: a novel networking paradigm for the collaborative community. "`https://archive-ouverte.unige.ch/unige:114607`", 2019.

[117] Thomas Kurian. Google cloud unveils strategy for telecommunications industry. "`https://cloud.google.com/blog/topics/inside-google-cloud/google-cloud-unveils-strategy-telecommunications-industry`", 2020.

[118] Daniel Larimer. Delegated proof-of-stake (dpos). *Bitshare whitepaper*, 2014.

[119] Ang Li, Xiaowei Yang, Srikanth Kandula, and Ming Zhang. Cloudcmp: comparing public cloud providers. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pages 1–14, 2010.

[120] S. Liu, L. Liu, J. Tang, B. Yu, Y. Wang, and W. Shi. Edge computing for autonomous driving: Opportunities and challenges. *Proceedings of the IEEE*, 107(8):1697–1716, Aug 2019.

[121] Tai Liu, Zain Tariq, Jay Chen, and Barath Raghavan. The barriers to overthrowing internet feudalism. In *Proceedings of the 16th ACM Workshop on Hot Topics in Networks*, pages 72–79, 2017.

[122] Wen-Chih Lo, Chih-Yuan Huang, and Cheng-Hsin Hsu. Edge-assisted rendering of 360 videos streamed to head-mounted virtual reality. In *2018 IEEE International Symposium on Multimedia (ISM)*, pages 44–51. IEEE, 2018.

[123] Yuan Lu, Qiang Tang, and Guiling Wang. Zebralancer: Private and anonymous crowdsourcing system atop open blockchain. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, pages 853–865. IEEE, 2018.

[124] Lele Ma, Shanhe Yi, and Qun Li. Efficient service handoff across edge servers via docker container migration. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, page 11. ACM, 2017.

[125] Andrew Machen, Shiqiang Wang, Kin K Leung, Bong Jun Ko, and Theodoros Salonidis. Live service migration in mobile edge clouds. *IEEE Wireless Communications*, 25(1):140–147, 2017.

[126] Dahlia Malkhi. Blockchain in the Lens of BFT. "`https://www.usenix.org/conference/atc18/presentation/malkhi`", 2018.

[127] Linux man page. Traceroute. "`https://linux.die.net/man/8/traceroute`", 2019.

[128] Katerina Mania, Bernard D Adelstein, Stephen R Ellis, and Michael I Hill. Perceptual sensitivity to head tracking latency in virtual environments with varying degrees of scene complexity. In *Proceedings of the 1st Symposium on Applied perception in graphics and visualization*, pages 39–47, 2004.

[129] Yuyi Mao, Changsheng You, Jun Zhang, Kaibin Huang, and Khaled B Letaief. A survey on mobile edge computing: The communication perspective. *IEEE Communications Surveys & Tutorials*, 19(4):2322–2358, 2017.

[130] R Timothy Marler and Jasbir S Arora. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26(6):369–395, 2004.

[131] R Timothy Marler and Jasbir S Arora. The weighted sum method for multi-objective optimization: new insights. *Structural and multidisciplinary optimization*, 41(6):853–862, 2010.

[132] R Preston McAfee. A dominant strategy double auction. *Journal of economic Theory*, 56(2):434–450, 1992.

[133] Microsoft. Azure functions. "`https://azure.microsoft.com/en-gb/services/functions/`", 2020.

[134] Microsoft. Azure regions. "https://azure.microsoft.com/en-us/global-infrastructure/regions/", 2020.

[135] Microsoft. Azure stack. "https://azure.microsoft.com/en-gb/overview/azure-stack/", 2020.

[136] Microsoft. The future of computing: intelligent cloud and intelligent edge. "https://azure.microsoft.com/en-gb/overview/future-of-cloud/", 2020.

[137] Nitinder Mohan. Edge computing platforms and protocols. "http://urn.fi/URN:ISBN:978-951-51-5561-0", 2019.

[138] Nitinder Mohan and Jussi Kangasharju. Edge-fog cloud: A distributed cloud for internet of things computations. In *2016 Cloudification of the Internet of Things (CIoT)*, pages 1–6, Nov 2016.

[139] Nitinder Mohan, Aleksandr Zavodovski, Pengyuan Zhou, and Jussi Kangasharju. Anveshak: Placing edge servers in the wild. In *Proceedings of the 2018 Workshop on Mobile Edge Communications*, MECOMM'18, pages 7–12, New York, NY, USA, 2018. ACM.

[140] Nitinder Mohan, Pengyuan Zhou, Keerthana Govindaraj, and Jussi Kangasharju. Grouping computational data in resource caches of edge-fog cloud. In *Proceedings of the 4th Workshop on CrossCloud Infrastructures and Platforms*, Crosscloud17, New York, NY, USA, 2017. Association for Computing Machinery.

[141] Roberto Morabito, Vittorio Cozzolino, Aaron Yi Ding, Nicklas Beijar, and Jorg Ott. Consolidate iot edge computing with lightweight virtualization. *IEEE Network*, 32(1):102–111, 2018.

[142] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.

[143] Arvind Narayanan, Eman Ramadan, Jason Carpenter, Qingxu Liu, Yu Liu, Feng Qian, and Zhi-Li Zhang. A first look at commercial 5g performance on smartphones. In *Proceedings of The Web Conference 2020*, WWW 20, page 894905, New York, NY, USA, 2020. Association for Computing Machinery.

[144] Stefan Nastic, Thomas Rausch, Ognjen Scekic, Schahram Dustdar, Marjan Gusev, Bojana Koteska, Magdalena Kostoska, Boro Jakimovski, Sasko Ristov, and Radu Prodan. A serverless real-time data

analytics platform for edge computing. *IEEE Internet Computing*, 21(4):64–71, 2017.

[145] Cloud Native. Kube edge. "`https://kubeedge.io`", 2020.

[146] Cloud Native. Kubernetes. "`https://kubernetes.io/`", 2020.

[147] Sambit Nayak, Nanjangud C Narendra, Anshu Shukla, and James Kempf. Saranyu: Using smart contracts and blockchain for cloud tenant management. In *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, pages 857–861. IEEE, 2018.

[148] Anselme Ndikumana, Nguyen H Tran, Ki Tae Kim, Choong Seon Hong, et al. Deep learning based caching for self-driving cars in multi-access edge computing. *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[149] OpenFog Consortium Architecture Working Group. OpenFog Reference Architecture for Fog Computing. "`https://www.iiconsortium.org/pdf/OpenFog_Reference_Architecture_2_09_17.pdf"`", 2017.

[150] OpenNebula. Opennebula technology. "`http://opennebula.org/`", 2018.

[151] OpenStack. Openstack technology. "`http://www.openstack.org/`", 2018.

[152] Claus Pahl and Brian Lee. Containers and clusters for edge cloud architectures–a technology review. In *2015 3rd international conference on future internet of things and cloud*, pages 379–386. IEEE, 2015.

[153] J. Pan and J. McElhannon. Future edge cloud and edge computing for internet of things applications. *IEEE Internet of Things Journal*, 5(1):439–449, Feb 2018.

[154] Rina Panigrahy et al. Heuristics for vector bin packing. *Microsoft Research*, 2011.

[155] V Pareto and Manuale di Economia Politica. Societa editrice libraria, milano, italy, 1906. translated into english by as schwier as manual of political economy, 1971.

[156] Stefan Perlebach. Golem network review and tutorial. "https://medium.com/trusteddapps/dapps2go-golem-network-1171336da51f", June 2018.

[157] Dana Petcu. Multi-cloud: expectations and current approaches. In *Proceedings of the 2013 international workshop on Multi-cloud applications and federated clouds*, pages 1–6, 2013.

[158] Larry Peterson, Tom Anderson, Sachin Katti, Nick McKeown, Guru Parulkar, Jennifer Rexford, Mahadev Satyanarayanan, Oguz Sunay, and Amin Vahdat. Democratizing the network edge. *ACM SIGCOMM Computer Communication Review*, 49(2):31–36, 2019.

[159] G. Premsankar, M. Di Francesco, and T. Taleb. Edge computing for the internet of things: A case study. *IEEE Internet of Things Journal*, 5(2):1275–1284, April 2018.

[160] DERO project. Dero Project - The World's Fastest Anonymous Blockchain. "https://dero.io/", 2020.

[161] Kjetil Raaen, Ragnhild Eg, and Carsten Griwodz. Can gamers detect cloud delay? In *2014 13th Annual Workshop on Network and Systems Support for Games*, pages 1–3. IEEE, 2014.

[162] Md Abdur Rahman, Md Mamunur Rashid, M Shamim Hossain, Elham Hassanain, Mohammed F Alhamid, and Mohsen Guizani. Blockchain and iot-based cognitive edge framework for sharing economy services in a smart city. *IEEE Access*, 7:18611–18621, 2019.

[163] Amir M Rahmani, Pasi Liljeberg, Jürgo-Sören Preden, and Axel Jantsch. *Fog computing in the internet of things: Intelligence at the edge*. Springer, 2017.

[164] Thomas Rausch, Waldemar Hummer, Vinod Muthusamy, Alexander Rashed, and Schahram Dustdar. Towards a serverless platform for edge {AI}. In *2nd {USENIX} Workshop on Hot Topics in Edge Computing (HotEdge 19)*, 2019.

[165] Siraj Raval. *Decentralized Applications: Harnessing Bitcoin's Blockchain Technology*. O'Reilly Media, Inc., 2016.

[166] Rethink Research. Cisco pushes IoT analytics to the extreme edge with mist computing. "https://rethinkresearch.biz/articles/cisco-pushes-iot-analytics-extreme-edge-mist-computing-2/"", 2014.

[167] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies. The Case for VM-Based Cloudlets in Mobile Computing. *IEEE Pervasive Computing*, 8(4):14–23, 2009.

[168] Mahadev Satyanarayanan. The emergence of edge computing. *Computer*, 50(1):30–39, 2017.

[169] Mahadev Satyanarayanan, Pieter Simoens, Yu Xiao, Padmanabhan Pillai, Zhuo Chen, Kiryong Ha, Wenlu Hu, and Brandon Amos. Edge analytics in the internet of things. *IEEE Pervasive Computing*, 14(2):24–31, 2015.

[170] Eder J Scheid, Bruno B Rodrigues, Lisandro Z Granville, and Burkhard Stiller. Enabling dynamic sla compensation using blockchain-based smart contracts. In *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pages 53–61. IEEE, 2019.

[171] Eder John Scheid and Burkhard Stiller. Leveraging smart contracts for automatic sla compensation-the case of nfv environment. In *IFIP 12th International Conference on Autonomous Infrastructure, Management and Security, AIMS*, pages 70–74, 2018.

[172] Michael Schneider, Jason Rambach, and Didier Stricker. Augmented reality based on edge computing using the example of remote live support. In *2017 IEEE International Conference on Industrial Technology (ICIT)*, pages 1277–1282. IEEE, 2017.

[173] Vincenzo Scoca, Rafael Brundo Uriarte, and Rocco De Nicola. Smart contract negotiation in cloud computing. In *2017 IEEE 10th International Conference on Cloud Computing (CLOUD)*, pages 592–599. IEEE, 2017.

[174] P. K. Sharma, M. Chen, and J. H. Park. A software defined fog node based distributed blockchain cloud architecture for iot. *IEEE Access*, 6:115–124, 2018.

[175] W. Shi and S. Dustdar. The promise of edge computing. *Computer*, 49(5):78–81, May 2016.

[176] Pedro Silva, Christian Perez, and Frédéric Desprez. Efficient heuristics for placing large-scale distributed applications on multiple clouds. In *2016 16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, pages 483–492. IEEE, 2016.

[177] Alessio Silvestro, Nitinder Mohan, Jussi Kangasharju, Fabian Schneider, and Xiaoming Fu. MUTE: Multi-tier edge networks. In *Proceedings of the 5th Workshop on CrossCloud Infrastructures & Platforms*, pages 1–7. ACM, 2018.

[178] Karolj Skala, Davor Davidovic, Enis Afgan, Ivan Sovic, and Zorislav Sojat. Scalable distributed computing hierarchy: Cloud, fog and dew computing. *Open Journal of Cloud Computing (OJCC)*, 2(1):16–24, 2015.

[179] Sonm. "`https://sonm.com/`", 2020.

[180] Alexandru Stanciu. Blockchain based distributed control system for edge computing. In *2017 21st International Conference on Control Systems and Computer Science (CSCS)*, pages 667–671. IEEE, 2017.

[181] Nick Szabo. Smart Contracts: Building Blocks for Digital Markets. "`http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_2.html`", 1996.

[182] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella. On multi-access edge computing: A survey of the emerging 5g network edge cloud architecture and orchestration. *IEEE Communications Surveys Tutorials*, 19(3):1657–1681, 2017.

[183] Tarik Taleb, Sunny Dutta, Adlen Ksentini, Muddesar Iqbal, and Hannu Flinck. Mobile edge computing potential in making cities smarter. *IEEE Communications Magazine*, 55(3):38–43, 2017.

[184] Noshina Tariq, Muhammad Asim, Feras Al-Obeidat, Muhammad Zubair Farooqi, Thar Baker, Mohammad Hammoudeh, and Ibrahim Ghafir. The security of big data in fog-enabled iot applications including blockchain: a survey. *Sensors*, 19(8):1788, 2019.

[185] Zahra Tarkhani, Anil Madhavapeddy, and Richard Mortier. Snape: The dark art of handling heterogeneous enclaves. In *Proceedings of the 2nd International Workshop on Edge Systems, Analytics and Networking*, EdgeSys 19, page 4853, New York, NY, USA, 2019. Association for Computing Machinery.

[186] Tether. "`https://tether.to`", 2020.

[187] The DFINITY. "`https://dfinity.org`", 2020.

[188] Rafael Brundo Uriarte, Rocco De Nicola, and Kyriakos Kritikos. Towards distributed sla management with smart contracts and blockchain. In *2018 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 266–271. IEEE, 2018.

[189] William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of finance*, 16(1):8–37, 1961.

[190] Paul Vigna and Steven Russolillo. The Mystery Behind Tether, the Crypto Worlds Digital Dollar. *The Wall Street Journal*, 2018. `https://www.wsj.com/articles/the-mystery-behind-tether-the-crypto-worlds-digital-dollar-1534089601`.

[191] Massimo Villari, Maria Fazio, Schahram Dustdar, Omer Rana, and Rajiv Ranjan. Osmotic computing: A new paradigm for edge/cloud integration. *IEEE Cloud Computing*, 3(6):76–83, 2016.

[192] WebAssembly. "`https://webassembly.org/`", 2020.

[193] Joe Weinman. *Cloudonomics: The business value of cloud computing*. John Wiley & Sons, 2012.

[194] Julia White. Microsoft azure enables a new wave of edge computing. heres how. "`https://azure.microsoft.com/en-gb/blog/microsoft-azure-enables-a-new-wave-of-edge-computing-here-s-how/`", 2018.

[195] Wikipedia. 5g. "`https://en.wikipedia.org/wiki/5G/`", 2020.

[196] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper*, 2014. `http://gavwood.com/paper.pdf`.

[197] David L Woods, John M Wyma, E William Yund, Timothy J Herron, and Bruce Reed. Factors influencing the latency of simple reaction time. *Frontiers in human neuroscience*, 9:131, 2015.

[198] Kwame-Lante Wright, Martin Martinez, Uday Chadha, and Bhaskar Krishnamachari. Smartedge: a smart contract for edge computing. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 1685–1690. IEEE, 2018.

[199] Xen. Xen hypervisor. "`http://www.xen.org`", 2018.

[200] Brent Xu, Dhruv Luthra, Zak Cole, and Nate Blakely. EOS: An Architectural, Performance, and Economic Analysis. "`https://www.whiteblock.io/library/eos-test-report.pdf`", 2018.

[201] Xiaolong Xu, Qing Cai, Guoming Zhang, Jie Zhang, Wei Tian, Xiaorui Zhang, and Alex X Liu. An incentive mechanism for crowdsourcing markets with social welfare maximization in cloud-edge computing. *Concurrency and Computation: Practice and Experience*, 2018.

[202] Jian Yang, Zhihui Lu, and Jie Wu. Smart-toy-edge-computing-oriented data exchange based on blockchain. *Journal of Systems Architecture*, 87:36–48, 2018.

[203] Ruizhe Yang, F Richard Yu, Pengbo Si, Zhaoxin Yang, and Yanhua Zhang. Integrated blockchain and edge computing systems: A survey, some research issues and challenges. *IEEE Communications Surveys & Tutorials*, 21(2):1508–1532, 2019.

[204] Shanhe Yi, Zijiang Hao, Qingyang Zhang, Quan Zhang, Weisong Shi, and Qun Li. Lavea: Latency-aware video analytics on edge computing platform. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, pages 1–13, 2017.

[205] Matei Zaharia, Benjamin Hindman, Andy Konwinski, Ali Ghodsi, Anthony D Joseph, Randy H Katz, Scott Shenker, and Ion Stoica. The datacenter needs an operating system. In *HotCloud*, 2011.

[206] A. Zavodovski, S. Bayhan, N. Mohan, P. Zhou, W. Wong, and J. Kangasharju. DeCloud: Truthful Decentralized Double Auction for Edge Clouds. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pages 2157–2167, 2019.

[207] A. Zavodovski, S. Bayhan, N. Mohan, P. Zhou, W. Wong, and J. Kangasharju. Decloud: Truthful decentralized double auction for edge clouds. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pages 2157–2167, 2019.

[208] Aleksandr Zavodovski, Nitinder Mohan, Suzan Bayhan, Walter Wong, and Jussi Kangasharju. ICON: Intelligent container overlays. In *Proceedings of the 17th ACM Workshop on Hot Topics in Networks*, HotNets '18, pages 15–21, New York, NY, USA, 2018. ACM.

[209] Aleksandr Zavodovski, Nitinder Mohan, Suzan Bayhan, Walter Wong, and Jussi Kangasharju. ExEC: Elastic extensible edge cloud. In *Proceedings of the 2Nd International Workshop on Edge Systems, Analytics and Networking*, EdgeSys '19, pages 24–29, New York, NY, USA, 2019. ACM.

[210] Aleksandr Zavodovski, Nitinder Mohan, Walter Wong, and Jussi Kangasharju. Open infrastructure for edge: A distributed ledger outlook. In *2nd {USENIX} Workshop on Hot Topics in Edge Computing (HotEdge 19)*, 2019.

[211] Lei Zhao, Jiajia Liu, Yongpeng Shi, Wen Sun, and Hongzhi Guo. Optimal placement of virtual machines in mobile edge computing. In *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pages 1–6. IEEE, 2017.

[212] Yang Zhao, Jun Zhao, Linshan Jiang, Rui Tan, and Dusit Niyato. Mobile edge computing, blockchain and reputation-based crowdsourcing iot federated learning: A secure, decentralized and privacy-preserving system, 2019.

[213] Liang Zheng et al. How to bid the cloud. *SIGCOMM Comput. Commun. Rev.*, 45(4), 2015.

[214] Huan Zhou, Cees de Laat, and Zhiming Zhao. Trustworthy cloud service level agreement enforcement with blockchain based smart contract. In *2018 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 255–260. IEEE, 2018.

[215] Huan Zhou, Xue Ouyang, Zhijie Ren, Jinshu Su, Cees de Laat, and Zhiming Zhao. A blockchain based witness model for trustworthy cloud service level agreement enforcement. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 1567–1575. IEEE, 2019.

[216] P. Zhou, T. Braud, A. Alhilal, P. Hui, and J. Kangasharju. Erl: Edge based reinforcement learning for optimized urban traffic light control. In *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 849–854, March 2019.

[217] Pengyuan Zhou, Wenxiao Zhang, Tristan Braud, Pan Hui, and Jussi Kangasharju. Arve: Augmented reality applications in vehicle to

edge networks. In *Proceedings of the 2018 Workshop on Mobile Edge Communications*, pages 25–30, 2018.